CENTRO UNIVERSITÁRIO FEI
JOÃO BATISTA DA SILVA JÚNIOR

PROJETO E CARACTERIZAÇÃO DE DISCRIMINADOR DE CORES UTILIZANDO FOTODIODO PIN LATERAL COM PORTA ATRAVÉS DE APRENDIZADO DE MÁQUINA

São Bernardo do Campo
2021

JOÃO BATISTA DA SILVA JÚNIOR

PROJETO E CARACTERIZAÇÃO DE DISCRIMINADOR DE CORES UTILIZANDO FOTODIODO PIN LATERAL COM PORTA ATRAVÉS DE APRENDIZADO DE MÁQUINA

Tese apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Doutor em Engenharia Elétrica. Orientado pelo Prof. Dr. Renato Camargo Giacomini.

São Bernardo do Campo

2021

da Silva Júnior, João Batista.
PROJETO E CARACTERIZAÇÃO DE DISCRIMINADOR DE CORES UTILIZANDO FOTODIODO PIN LATERAL COM PORTA ATRAVÉS DO APRENDIZADO DE MÁQUINA / João Batista da Silva Júnior. São Bernardo do Campo, 2021.
132 p. : il.

Tese - Centro Universitário FEI.
Orientador: Prof. Dr. Renato Camargo Giacomini.

1. Fotodiodo PIN. 2. Polarização de Porta VG. 3. Aprendizado de Máquina. I. Giacomini, Renato Camargo, orient. II. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os dados fornecidos pelo(a) autor(a).

centro universitário FEI

APRESENTAÇÃO DE TESE
ATA DA BANCA EXAMINADORA

Programa de Pós-Graduação Stricto Sensu em Engenharia Elétrica

Doutorado

PGE-10

Aluno: João Batista da Silva Júnior **Matrícula:** 515306-9

Título do Trabalho: PROJETO E CARACTERIZAÇÃO DE DISCRIMINARDOR DE CORES UTILIZANDO FOTODIODO PIN LATERAL COM PORTA ATRAVÉS DE APRENDIZADO DE MÁQUINA.

Área de Concentração: Nanoeletrônica e Circuitos Integrados

Orientador: Prof. Dr. Renato Camargo Giacomini

Data da realização da defesa: 28/06/2021

ORIGINAL ASSINADA

Avaliação da Banca Examinadora

O candidato fez a apresentação inicial em cerca de 50 minutos, relatando a pesquisa, seus resultados e conclusão. Em seguida, foi arguido pelo banca examinadora. As dúvidas, críticas e sugestões foram apresentadas por parte de cada examinador e foi dada ao candidato a oportunidade de responder, esclarecer ou corrigir. A banca reuniu-se em sessão reservada e deliberou, aprovando o candidato e solicitando alterações para o trabalho final.

São Bernardo do Campo, / / .

MEMBROS DA BANCA EXAMINADORA	
Prof. Dr. Renato Camargo Giacomini	Ass.: _____
Prof. Dr. Vanderlei Cunha Parro	Ass.: _____
Prof. Dr. Carlos Eduardo Thomaz	Ass.: _____
Prof. Dr. Lucas Compassi Severo	Ass.: _____
Prof. Dr. John Paul Hempel Lima	Ass.: _____

A Banca Examinadora acima-assinada atribuiu ao aluno o seguinte:

APROVADO ☒ REPROVADO ☐

VERSÃO FINAL DA TESE

ENDOSSO DO ORIENTADOR APÓS A INCLUSÃO DAS RECOMENDAÇÕES DA BANCA EXAMINADORA

_____

Aprovação do Coordenador do Programa de Pós-graduação

_____

Prof. Dr. Carlos Eduardo Thomaz

Dedico este trabalho primeiramente a Deus, pela sua infinita misericórdia, à minha mãe Cícera, pai João Batista, irmãos Jacqueline, Jorge, à esposa Sarah e ao meu pequeno Gabriel pelo amor, compreensão, paciência e motivação durante esta grande jornada.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Ao senhor meu Deus por ter me apresentado sua graça e guiando-me em caminhos retos, ensinando os verdadeiros preceitos da vida.

Ao Prof. Dr. Renato Camargo Giacomini, meu orientador e amigo, por acreditar no meu potencial, sendo um grande consultor direcionando-me para os obstáculos que apareciam, propondo oportunidades de expandir meus conhecimentos neste novo caminho que irei trilhar.

Ao amigo Rudolf Bühler, que em todo o trabalho meu deu suporte necessário em momentos difíceis, pela sua generosidade, troca de experiências, grande amizade e motivação para alcançar os objetivos.

Aos amigos Arianne Pereira, Carla Dick, André Perin, Milene Galeti e Juliano Oliveira pela amizade de longa data, pela generosidade e ajuda em todos os momentos.

Aos professores Dr. Carlos Eduardo Thomaz e Dra. Michelly de Souza, pelos conhecimentos e orientações transmitidas durante a qualificação.

À minha mãe Cícera Gomes, ao meu pai João Batista, aos meus irmãos Antônio Jorge e Jacqueline Kevely, pela paciência, amor, companheirismo, conselhos, motivação e pelas condições que me deram oportunidade de ingressar no mundo corporativo.

À minha esposa Sarah Parschin, que esteve me ajudando em todos os momentos difíceis, pelo seu amor, companheirismo, compreensão e conselhos, sendo a minha muleta nos momentos mais difíceis da minha vida.

Ao meu filho Gabriel, que veio nos presentear este ano com seu nascimento e bênçãos para nossa família.

Aos meus sogros Paulo Parschin e Elisabeth Parschin, que me acolheram como filho, e aos meus cunhados André, Thiago, Samuel e Fernanda pela amizade e companheirismo em família.

Ao Centro Universitário da FEI, por oferecer toda infraestrutura necessária para alcançar os objetivos impostos neste trabalho, desde o período da graduação, seguido do mestrado e finalizando o ciclo no doutorado, sendo minha casa, vestindo sua camisa com orgulho, como um feiano.

As demais pessoas, colaboradores da FEI, colegas e amigos de trabalho, que me ajudaram e motivaram para realização deste trabalho de forma direta e indireta.

Sei bem o que é passar necessidades e sei o que é andar com fartura. Aprendi o mistério de viver feliz em todo lugar e em qualquer situação, esteja bem alimentado, ou mesmo com fome, possuindo fartura, ou possuindo privações. Tudo posso naquele que me fortalece.

Carta de Paulo aos Filipenses 4:12 ao 13

RESUMO

O avanço tecnológico no mercado de sensores de imagem vem crescendo de forma ao longo dos anos com buscas incessantes na detecção da radiação eletromagnética, nas áreas de segurança, medicina, aeroespacial, câmeras fotográficas, televisões, celulares etc. Essa busca motivou a comunidade científica a realizar estudos dos fotodiodos PIN para detecção da radiação eletromagnética, que são dispositivos capazes de converter a energia luminosa em um sinal elétrico. Ainda existem grandes lacunas nos estudos dos fotodetectores, motivando pesquisadores do mundo inteiro a estudá-los. Além do mercado óptico, outra tecnologia imersiva está em ascensão, a IA (Inteligência Artificial), dividida em três grandes grupos: Aprendizado de Máquina (AM), Ciência de Dados e "*Big Data*", que permitirá entender e ensinar as tomadas de decisões das máquinas. No mercado de aquisição de imagens, as câmeras de hoje possuem boa resolução, porém, seu modelo de tratativa da informação ainda é limitado, devido à utilização do modelo RGB, que combinas as radiações R (*RED* – Vermelho) G (*GREEN* – Verde) e B (*BLUE* – Azul) para gerar as demais radiações do espectro eletromagnético visível. A captura dessas radiações ocorre através da detecção dos comprimentos de ondas via filtros ópticos discretos depositados na matriz dos fotodetectores. Em virtude das lacunas apresentadas nos estudos dos fotodetectores e da ascensão da IA, este trabalho visa estudar o fotodiodo PIN lateral com porta para discriminar as cores do espectro visível com auxílio dos algoritmos do aprendizado de máquina, no intuito de expandir a qualidade das imagens capitadas. As análises foram divididas em três partes sendo que na primeira parte, o estudo propõe índices de mérito capazes de discriminar as cores através da polarização de porta V_G. Os índices de mérito X_{RGB}, Y_{RGB}, $I'_{CAT}(V_G)$ e $I_{CAT,int}$ demostram essa possibilidade. Para os índices X_{RGB} e Y_{RGB} os estudos mostram que a faixa de V_G se estende de -0,8V à +0,8V. Já para o método derivativo ($I'_{CAT}(V_G)$) a faixa de V_G para discriminar está compreendida entre -0,8V à -0,5V, tendo o ponto máximo de discriminação para V_G = -0,65V. Finalizando os estudos dos índices, o método integrativo ($I_{CAT,int}(V_G)$) informa que a melhor faixa para se trabalhar é 0V à 0,5V, onde o comportamento é linear, apresentando como discriminador os coeficientes angulares (B_X) de cada comprimento de onda RGB. Na segunda parte, é realizado o estudo dos algoritmos gerados para os dados caracterizados do fotodiodo PIN lateral com porta, sendo: *Linear Discriminant*, *Quadratic Discriminant*, *Linear SVM*, *Quadratic SVM*, *Cubic SVM*, *Cosine KNN* e *Ensemble Subspace Discriminant*. Todos os modelos durante a fase de treinamento mostraram acurácia acima de 90% e ROC/AUC = 1. Na fase de teste, todos os modelos ficaram com acurácia acima de 94%, sendo que, três deles,

Linear Discriminant, *Quadratic Discriminant* e *Quadratic SVM* obtiveram 100% de acurácia. Na terceira parte, é proposta a discriminação através da modificação do um circuito eletrônico discutido no item 3.6 deste trabalho, através das polarizações V_G que são quatro no total, -1,0 V, -0,6 V, +0,6 V e +1,0 V. Esse circuito fornece a relação entre a tensão de saída V_λ pelo comprimento de onda λ, através da razão logarítmica entre as correntes dos fotodiodos laterais com porta. A melhor resposta de detecção foi apresentada para V_{G1} = +1,0 V e V_{G2} = -0,6 V, para as três potências ópticas em análises (0,1; 0,5 e 1,0 W/m^2). As demais combinações de V_G demostram resultados satisfatórios para qualquer valor de potência óptica para a faixa de λ entre 200 nm a 500nm, onde existe uma boa variação na tensão de saída (V_λ) do circuito. O terminal de porta proposto neste dispositivo auxilia positivamente na discriminação dos comprimentos de ondas, abrindo novos horizontes na detecção da radiação eletromagnética.

Palavras-chave: Fotodiodo PIN. Polarização de Porta V_G. Aprendizado de Máquina.

ABSTRACT

Technological advancement in the imaging market in general has been growing exorbitantly over the years with relentless searches in the detection of electromagnetic radiation, in the areas of security, medicine, aerospace, cameras, televisions, cell phones and etc. This search motivated the scientific community to carry out studies of PIN photodiodes for detecting electromagnetic radiation, which are devices capable of converting light energy into an electrical signal. There are still great gaps in the studies of photodetectors, greatly motivating researchers from all over the world to study them. In addition to the optical market, another immersive technology is on the rise, the AI (Artificial Intelligence), divided into three major groups: Machine Learning (AM), Data Science and "Big Data", which will allow understanding and teaching of machine decisions. In the image acquisition market, today's cameras have good resolution, however, their model for dealing with information is still limited, due to the use of the RGB model, which combines the primary radiation R (RED - Red) G (GREEN - Green) and B (BLUE - Blue) to generate the other radiations of the visible electromagnetic spectrum. The acquisition of these colors occurs through the detection of wavelengths via discrete optical filters deposited in the photodetector matrix. Due to the gaps presented in the studies of photodetectors and the rise of AI, this work aims to study the lateral PIN gatede photodiode to discriminate the colors of the visible spectrum with the help of machine learning algorithms, in order to expand the quality of the capitulated images. The analyzes were divided into three parts and the first part, the study proposes indications of merits capable of discriminating colors through the gate polarization (V_G). The merit indexes X_{RGB}, Y_{RGB}, I'_{CAT} (V_G) and $_{ICAT,\ int}$ demonstrate this possibility. For the X_{RGB} and Y_{RGB} indices, studies show that the V_G range extends from -0.8V to + 0.8V. For the derivative method (I'_{CAT} (V_G)), the range of V_G to discriminate is between -0.8V to -0.5V, with the maximum point of discrimination for V_G = -0.65V. Finishing the studies of the indexes, the integrative method ($I_{CAT,\ int}$ (V_G)) informs that the best range to work on is 0V to 0.5V, where the behavior is linear, presenting the angular coefficients (BX) of each length as discriminator RGB waveform. In the second part, the study of the algorithms generated for the characterized data of the lateral PIN gated photodiode is carried out, being: Linear Discriminant, Quadratic Discriminant, Linear SVM, Quadratic SVM, Cubic SVM, Cosine KNN and Ensemble Subspace Discriminant. All models during the training phase showed accuracy above 90% and ROC / AUC = 1. In the test phase, all models were accurate above 94%, with three of them, Linear Discriminant, Quadratic Discriminant and Quadratic SVM obtained 100% accuracy. In the third part, discrimination is proposed through

the modification of an electronic circuit discussed in item 3.6 of this work, through the V_G polarizations which are four in total, -1.0 V, -0.6 V, +0.6 V and +1.0 V. This circuit provides the relationship between the output voltage V_λ by the wavelength λ, through the logarithmic ratio between the currents of the lateral photodiodes with port. The best detection response was presented for V_{G1} = +1.0 V and V_{G2} = -0.6 V, for the three optical powers in analysis (0.1; 0.5 and 1.0 W / m^2). The other V_G combinations show satisfactory results for any value of optical power for the λ range between 200 nm to 500nm, where there is a good variation in the output voltage (V_λ) of the circuit. The gate terminal proposed in this device positively assists in the discrimination of wavelengths, opening new horizons in the detection of electromagnetic radiation.

Keywords: PIN photodiode. Gate Polarization V_G. Machine Learning.

LISTA DE ILUSTRAÇÕES

Figura 1 – Níveis de energia dentro da faixa proibida. 23
Figura 2 – Transição por foto excitação direto e indireto. 24
Figura 3 – Coeficiente de absorção (α). 25
Figura 4 – Profundidade de penetração do Silício. 25
Figura 5 – Fotodiodo PIN. 27
Figura 6 - Tipos básicos de fotodiodos em tecnologia CMOS: P+/Nwell, N+/PSUB e Nwell/PSUB. 29
Figura 7 – Fotodiodo PIN vertical. 30
Figura 8 – Fotodiodo multi-dedo em Nwell em tecnologia CMOS convencional de 0,35µm. 30
Figura 9 – Representação do olho humano. 33
Figura 10 – Sensibilidade dos três tipos de cones da retina. 33
Figura 11 – Diagrama cromático do espaço de cores no CIE 1931. 34
Figura 12 – Representação do sistema aditivo de cores. 34
Figura 13 – Representação do sistema aditivo de cores. 35
Figura 14 – Representação do sistema aditivo de cores. 35
Figura 15 – Cubo do modelo de cores RGB. 36
Figura 16 – Filtro discreto CFA ou CFM. 38
Figura 17 – Sensor de imagem CCD. 39
Figura 18 – Sensor de imagem de pixel ativo. 40
Figura 19 – Sinal de saída do pixel ativo com e sem incidência de luminosidade. 41
Figura 20 – Sensor *Foveon X3*. 42
Figura 21 – Sensor TFD. 43
Figura 22 – A) Linhas de campo elétrico do sensor TFD, B) Estrutura do sensor TFD. 44
Figura 23 – Fotodetector vertical com perfil P^+-N-N proposto por (ASO; KAWANABE, 1982). 45
Figura 24 – Responsividade do fotodetector vertical proposto por (ASO; KAWANABE, 1982) 45
Figura 25 – Sistema de detecção de comprimento de onda e intensidade luminosa proposto por (ASO; KAWANABE, 1982). 46
Figura 26 – Vout vs λ do circuito subtrator. 47
Figura 27 – Estrutura básica do aprendizado de máquina. 49
Figura 28 – Fluxo para gerar um modelo computacional baseado no AM. 50

Figura 29 – Conjunto de dados para treinar um modelo no formato classe, atributo e valor...51
Figura 30 – Algoritmos usados no Aprendizado Supervisionado. 51
Figura 31 – Algoritmos usados no Aprendizado Não-Supervisionado. 52
Figura 32 – Matriz de confusão. 53
Figura 33 – Curva ROC. 55
Figura 34 – Curva AUC. 56
Figura 35 – Fotos dos equipamentos disponíveis na sala de caracterização elétrica do Centro Universitário da FEI: (A) Analisador de parâmetros Keithley 4200, (B) Microprovador Cascade Microtech REL 3600. 57
Figura 36 – Esquema de ligação para caracterização da potência óptica incidente dos LEDs. 58
Figura 37 – Esquema de ligação para caracterização da tensão e corrente elétrica dos LEDs. 58
Figura 38 – Suporte para minimizar o espalhamento da luz dos Leds. 59
Figura 39 – Esquema do fotodiodo PIN lateral com porta com 5 dedos. 64
Figura 40 – Taxa da área fotossensível vs comprimento intrínseco. 65
Figura 41 – Leiaute real do fotodiodo PIN lateral com porta com 5 dedos. 65
Figura 42 – Corrente de catodo (I_{CAT}) pela tensão de porta (V_G) entre -0,8V até 0,8V com polarização reversa de 0,45V (V_{CAT}) (experimental). 66
Figura 43 – Corrente de catodo pela tensão de porta entre -0,8V até 0,8V com polarização reversa de 0,45V comportamento linear (experimental). 67
Figura 44 – Eficiência Quântica Total (EQ_T) em função da tensão de porta para potência de 0,6 W/m^2 (experimental). 68
Figura 45 – Relação Sinal Ruído (SNR) em função da tensão de porta para a potência de 0,6 W/m^2 (experimental). 69
Figura 46 – Corrente de catodo (I_{CAT}) em função do comprimento de onda (λ) (experimental + simulado). 70
Figura 47 – Eficiência Quântica Total (EQ_T) em função do comprimento de onda (λ) (experimental + simulado). 70
Figura 48 – $X_{RGB\ (AZUL)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental). 71
Figura 49 – $X_{RGB\ (VERDE)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental). 72
Figura 50 – $X_{RGB\ (VERMELHO)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental). ...72
Figura 51 – Região de depleção vertical e lateral do fotodiodo, ...74
Figura 52 – Gráfico de barras X_{RGB} para V_G=+0,2V (experimental). 74
Figura 53 – Y_{RGB} vs V_G entre -0.8 até 0.6V (experimental).....75
Figura 54 – Gráfico de barras Y_{RGB} para V_G=+0,2V (experimental). 77

Figura 55 – Gráfico de $I'_{CAT}(V_G)$ x V_G para V_{CAT} = 0,45V (experimental). 80
Figura 56 – Gráfico expandido da faixa -0,8V à -0,5V de $I'_{CAT}(V_G)$ x V_G para V_{CAT} = 0,45V (experimental). 81
Figura 57 – Gráfico de $I'_{CAT}(V_G)$ x V_G para V_G = 0,65V (experimental). 82
Figura 58 – Diagrama de blocos do método derivacional para discriminação RGB. 82
Figura 59 – Gráfico de $I_{CAT,int}(V_G)$ x V_G para V_{CAT} = 0,45V (experimental). 83
Figura 60 – Gráfico de $I_{CAT,int}(V_G)$ x V_G para faixa de V_G de 0V à 0,5V (experimental). 84
Figura 61 – Gráfico de Coeficiente angular vs Potência do método integrativo (experimental). 86
Figura 62 – Diagrama de blocos do método integrativo para discriminação RGB. 86
Figura 63 – Diagrama de blocos de todas as métricas para discriminação das cores RGB. 87
Figura 64 – Gráfico dos dados V_G = -0,6V pelo V_G = -1,0 V para faixa de potência de 0,1 a 1,0 W/m^2 (Experimental + Simulado). 89
Figura 65 – Gráficos ROC/ACU. 93
Figura 66 – Quadro comparativo das médias das acurácias entre os modelos durante a fase de treinamento. 101
Figura 67 – Quadro comparativo das médias das acurácias entre os modelos durante a fase de teste. 101
Figura 68 – Diagrama de blocos dos sete algoritmos do aprendizado de máquina para classificar as cores RGB. 102
Figura 69 – Circuito eletrônico para detecção dos comprimentos de onda. 103
Figura 70 – I_{CAT} vs λ com P = 0,5 W/m^2 para as quatro tensões V_G usados nos algoritmos do AM. 105
Figura 71 – V_λ vs λ com P = 0,1 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V. 106
Figura 72 – V_λ vs λ com P = 0,5 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V. 106
Figura 73 – V_λ vs λ com P = 1,0 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V. 107
Figura 74 – Comparativo V_λ vs λ com IF_{VG1} = -0,6 V / IF_{VG2} = -1,0 V para as três potências. 107
Figura 75 – Comparativo V_λ vs λ com IF_{VG1} = +1,0 V / IF_{VG2} = -1,0 V para as três potências. 108
Figura 76 – Comparativo V_λ vs λ com IF_{VG1} = +1,0 V / IF_{VG2} = -0,6 V para as três potências. 108

LISTA DE TABELAS

Tabela 1 – Dados caracterizados dos Leds (experimental). 60
Tabela 2 – Algoritmos oferecidos pelo *Classification Learner*. 61
Tabela 3 – Resultados da razão X_{RGB} para V_G=+0,2V (experimental). 73
Tabela 4 – Resultados da razão Y_{RGB} para V_G=+0,2V (experimental). 76
Tabela 5 – Aproximação polinomial para o azul (experimental). 78
Tabela 6 – Aproximação polinomial para o verde (experimental). 79
Tabela 7 – Aproximação polinomial para vermelho (experimental). 79
Tabela 8 – Dados extraídos do pico máximo da derivada (experimental). 81
Tabela 9 – Aproximação linear do azul (experimental). 85
Tabela 10 – Aproximação linear do verde (experimental). 85
Tabela 11 – Aproximação linear do vermelho (experimental). 85
Tabela 12 – Mapeamento dos dados I_{CAT} x V_G para faixa de potência de 0,1 à 1,0 W/m^2. 88
Tabela 13 – Resultados dos algoritmos durante a fase de treinamento (experimental + simulado). 89
Tabela 14 – Matriz de confusão do algoritmo *Linear Discriminant*. 91
Tabela 15 – Matriz de confusão do algoritmo *Quadratic Discriminant*. 91
Tabela 16 – Matriz de confusão do algoritmo *Linear SVM*. 91
Tabela 17 – Matriz de confusão do algoritmo *Quadratic SVM*. 91
Tabela 18 – Matriz de confusão do algoritmo *Cubic SVM*. 92
Tabela 19 – Matriz de confusão do algoritmo *Cosine KNN*. 92
Tabela 20 – Matriz de confusão do algoritmo *Ensemble Subspace Discriminant*. 92
Tabela 21 – Resumo do resultado ROC/AUC de cada algoritmo. 93
Tabela 22 – Resumo dos algoritmos durante a fase de treinamento. 94
Tabela 23 – Novos dado imputados para testar os algoritmos. 95
Tabela 24 – Matriz de confusão do algoritmo *Linear Discriminant*. 96
Tabela 25 – Matriz de confusão do algoritmo *Quadratic Discriminant*. 96
Tabela 26 – Matriz de confusão do algoritmo *Linear SVM*. 96
Tabela 27 – Matriz de confusão do algoritmo *Quadratic SVM*. 97
Tabela 28 – Matriz de confusão do algoritmo *Cubic SVM*. 97
Tabela 29 – Matriz de confusão do algoritmo *Cosine KNN*. 97
Tabela 30 – Matriz de confusão do algoritmo *Ensemble Subspace Discriminant*. 97
Tabela 31 – Resumo dos algoritmos durante a fase de teste. 98

Tabela 32 – Acurácia, média e desvio padrão para 7 rodadas da fase de treinamento.....99
Tabela 33 – Acurácia, média e desvio padrão para 7 rodadas da fase de teste. 100
Tabela 34 – Combinação dos valores de V_{G1} e V_{G2}. 105
Tabela 35 – Resumo comparativos das análises entre IF_{VG1} / IF_{VG2} para a tensão de saída V_{λ}. 109

LISTA DE SÍMBOLOS

b	Componente normalizada entre 0 e 1 da cor Azul (*Blue*)
b_A	Uma constante que intercepta I_{CAT},int(V_G)
bo	Valor que intercepta $I_{CAT}(V_G)$ que representa uma constante
B	Componente da cor Azul (*Blue*)
Bmáx	Intensidade máxima da componente da cor Azul (*Blue*)
B_X	Parâmetro da equação (coeficiente angular)
B1 à B9	Parâmetros da aproximação linear de I_{CAT} vs V_G
D_N	Constante de difusão dos elétrons [cm^2/s]
D_P	Constante de difusão das lacunas [cm^2/s]
E	Energia do fóton [eV]
E_a	Energia do nível permitido para impurezas aceitadoras dentro da faixa proibida [eV]
E_d	Energia do nível permitido para impurezas doadoras dentro da faixa proibida [eV]
EC	Energia da banda de condução [eV]
EQ_T	Eficiência quântica total de um fotodetector [adimensional]
EV	Energia da banda de valência [eV]
e-h	Par elétron-lácuna
f	Frequência [Hz]
g	Componente normalizada entre 0 e 1 da cor Verde (*Green*)
G	Componente da cor Verde (*Green*)
G_L	Taxa de geração de portadores para um dado comprimento de onda [$cm^{-3}s^{-1}$]
Gmáx	Intensidade máxima da componente da cor Verde (*Green*)
G_{TER}	Taxa de geração térmica [$cm^{-3}s^{-1}$]
h	Constante de Planck [$6.63x10^{-34}$J.s = $4,14x10^{-15}$eV.s]
I_{ESCURO}	Corrente de escuro [A]
I_{FG}	Corrente fotogerada [A]
I_{CAT}	Corrente de catodo do fotodiodo [A]
$I'_{CAT}(V_G)$	Derivada da corrente de catodo do fotodiodo pela sua tensão de porta [A/V]
$I_{CAT,int,linear}(V_G)$	Função da aproximação linear
IPD_1	Corrente coletada para comprimentos de ondas longos [A]
IDP_2	Corrente coletada para comprimentos de ondas curtos [A]

I_{F1}	Corrente do fotodiodo 1 [A]
I_{F2}	Corrente do fotodiodo 2 [A]
I_S	Corrente de saturação do diodo [A]
I_T	Corrente total do fotodiodo [A]
J_{ESCURO}	Densidade de corrente no escuro [A/m^2]
L_{ABS}	Profundidade de penetração [m]
L_i	Comprimento da região intrínseca do fotodiodo PIN`[m]
L_N	Comprimento de difusão dos elétrons [m]
L_P	Comprimento de difusão das lacunas [m]
k	Função de onda ou momento dos portadores [N.m]
K	Constante de Boltzman [1,38066 x 10^{-23} J/K]
L	Comprimento do transistor [µm]
N_A	Concentração de dopantes aceitadoras do tipo P [cm^{-3}]
N_D	Concentração de dopantes doadoras do tipo N [cm^{-3}]
n	Concentração de elétrons em um semicondutor intrínseco [cm^{-3}]
ni	Concentração intrínseca de portadores [cm^{-3}]
n_N	Concentração de elétrons num material tipo N [cm^{-3}]
n_P	Concentração de elétrons num material tipo P [cm^{-3}]
n_{P0}	Concentração de elétrons no equilíbrio dentro da região P [cm^{-3}]
p	Concentração de lacunas em um semicondutor intrínseco [cm^{-3}]
Pmáx.	Potência máxima do transitor [µW]
P_{IN}	Potência incidente [W/m^2]
p_N	Concentração de lacunas num material tipo N [cm^{-3}]
pn_0	Concentração de lacunas no equilíbrio dentro da região N [cm^{-3}]
p_P	Concentração de lacunas num material tipo P [cm^{-3}]
q	Carga elementar do elétron [1,6 x 10^{-19} C]
r	Componente normalizada entre 0 e 1 da cor Vermelha (*Red*)
R	Componente da cor Vermelha (*Red*)
R	Responsividade [A/W]
Rmáx	Intensidade máxima da componente da cor Vermelha (*Red*)
R_{TER}	Taxa de recombinação térmica [cm-^{3}s-^{1}]
SNR	Relação Sinal – Ruído [adimensional]
V_{A1}	Tensão de saída do amplificador operacional 1 [V]
V_{A2}	Tensão de saída do amplificador operacional 2 [V]

V_{CAT}	Tensão de polarização reversa do fotodiodo [V]
V_{CC}	Sinal de entrada do amplificador operacional [V]
V_{DD}	Tensão de alimentação [V]
V_F	Tensão direta do diodo quando polarizado [V]
V_G	Tensão de porta do fotodiodo [V]
V_{G1}	Faixa de tensão de porta 1 [V]
V_{G2}	Faixa de tensão de porta 2 [V]
Vout	Tensão de saída do circuito subtrator [V]
V_T	Tensão de limiar do transistor (*Threshold Voltage*) [V]
V_K	Tensão térmica [V]
V_λ	Tensão de saída do amplificador operacional 3 [V]
T	Temperatura em Kelvin [K]
Tc	Tempo de conversão [s]
X_{RGB1}	Razão entre a potência de 0,4 W/m^2 pela potência de 0,2 W/m^2 das cores vermelho, verde, azul
X_{RGB2}	Razão entre a potência de 0,6 W/m^2 pela potência de 0,2 W/m^2 das cores vermelho, verde, azul
X_{RGB3}	Razão entre a potência de 0,6 W/m^2 pela potência de 0,4 W/m^2 das cores vermelho, verde, azul
X_{RGB}	Razão entre as potências ópticas das cores vermelho, verde, azul
Y_{RGB}	Razão entre as cores vermelho, verde, azul
W	Comprimento da região de depleção [m]
W	Largura do transistor [µm]
α	Coeficiente de absorção [cm^{-1}]
λ	Comprimento de onda [nm]
λ_C	Maior comprimento de onda que pode ser absorvido pelo silício [m]
λ_{RGB}	Comprimento de onda RGB [nm]
τ_0	Tempo médio de vida das lacunas e dos elétrons livres [s]
ϕ	Intensidade de fluxo de fótons
ϕ_0	Intensidade de fluxo de fótons incidente a superfície do material

LISTA DE ABREVIATURAS

AM	Aprendizado de máquina
AEC1	Arquitetura Eletrônica Computacional 1
AEC2	Arquitetura Eletrônica Computacional 2
AOP	Amplificador Operacional
APS	Sensor de pixel ativo (*Active Pixel Sensor*)
BGN	*Band Gap Narrowing*
CCD	*Charged Couple Device*
CLK	Representa o pulso de *Clock*
CIE	*Commission Internationale de I'Eclairage* – Comissão Internacional de Iluminação
CMOS	*Complementary Metal Oxide Semiconductor*
CRT	Tudo de Raios Catódicos (*Cathode Ray Tube*)
D	Diodo de junção PN
DRAM	*Dynamic random-access memory*
FLDMOB	*Paralel Electric Field Dependece*
FPR	*False Positive Rate* (Taxa Falso Positivo)
IA	Inteligência Artificial
IOT	*Internet of Things*
KLAAUG	*Klaassen's concentration dependente Auger recombination model*
KNN	*k Nearest Neighbors*
LCD	Visor de Cristal Líquido (*Liquid Crystal Display*)
LED	Diodo Emissor de Luz (*Light Emitting Diode*)
MOS	*Metal Oxide Semiconductor*
MATLAB	*MAtrix LAboratory*
OLED	Diodo Orgânico Emissor de Luz (*Organic Light Emitting Diode*)
PD_1	Fotodiodo 1
PD_2	Fotodiodo 2
PPS	Sensor de pixel passivo (*Passive Pixel Sensor*)
PWM	*Pulse Width Modulation*
RDD	Região de Depleção
RGB	*Red-Green-Blue*
RS	Transistor Seguidor de Linha

SF	Transistor Seguidor de Fonte
SHI	*Shirahata*
SNR	Relação Sinal – Ruído (*Signal Noise Rate*)
SPAM	*Sending and Posting Advertisement in Mass*
SVM	*Support Vector Machine*
TFD	Sensor *Transverse Field Detectors*
FN	*False Negative* (Falso Negativo)
FP	*False Positive* (Falso Positivo)
TP	*True Positive* (Verdadeiro Positivo)
TPR	*True Positive Rate* (Taxa Verdadeiro Positivo
TN	*True Negative* (Verdadeiro Negativo)

SUMÁRIO

1 INTRODUÇÃO 19
1.1 MOTIVAÇÃO 20
1.2 ORGANIZAÇÃO DA TESE 21
2 CONCEITOS BÁSICOS SOBRE FOTOGERAÇÃO 22
2.1 FOTOGERAÇÃO E ABSORÇÃO 22
2.2 TIPOS DE FOTODETECTORES 27
2.2.1 Fotodiodo PIN 27
2.2.2 Tipos básicos de fotodiodos convencionais verticais 29
2.2.3 Fotodiodos PIN Multi-Dedos 30
2.3 DESEMPENHO DOS FOTODETECTORES 31
2.3.1 Responsividade 31
2.3.2 Relação sinal – ruído 31
2.3.3 Eficiência quântica 32
3 SENSORES ÓPTICOS EM TECNOLOGIA CCD E CMOS 32
3.1 COLORIMETRIA 32
3.1.1 Diagrama cromático CIE 34
3.1.2 Modelo de cores RGB e filtros CFA 36
3.2 SENSOR DE IMAGEM CCD 38
3.3 SENSOR DE IMAGEM CMOS 39
3.3.1 Sensor CMOS de pixel ativo (APS) 40
3.4 SENSOR ÓPTICO FOVEON X3 41
3.5 SENSOR ÓPTICO TFD 42
3.6 SISTEMA DE DETECÇÃO DE COMPRIMENTO DE ONDA E INTENSIDADE LUMINOSA 44
4 AMPRENDIZADO DE MÁQUINA 48
4.1 APRENDIZADO SUPERVISIONADO 50
4.2 APRENDIZADO NÃO-SUPERVISIONADO 52
4.3 MÉTRICAS DE DESEMPENHO EM APRENDIZADO DE MÁQUINA 52

4.3.1 Matriz de Confusão 53
4.3.2 Curva AUC/ROC 55
5 MATERIAIS E MÉTODOS 57
5.1 APARATO EXPERIMENTAL 57
5.1 O SIMULADOR NUMÉRICO MATLAB 60
5.2 SIMULADOR DE DISPOSITIVOS ELETRÔNICOS ATLAS 62
6 RESULTADOS EXPERIMENTAIS PARA O FOTODIODO PIN LATERAL COM PORTA 64
6.1 CARACTERIZAÇÃO DO FOTODIODO PIN LATERAL COM PORTA 66
6.2 INFLUÊNCIA DA POTÊNCIA ÓPTICA 71
6.3 USO DA DERIVADA E INTEGRAL PARA DISCRIMINAR AS CORES RGB 77
6.3.1 Método Derivacional 80
6.3.2 Método Integrativo 83
6.4 RESULTADOS PARCIAIS DA CARACTERIZAÇÃO DO FOTODIODO PIN LATERAL COM PORTA 87
7 RESULTADOS DOS MODELOS COMPUTACIONAIS GERADOS PELO APRENDIZADO DE MÁQUINA 88
7.4 RESULTADOS PARCIAIS DOS ALGORITMOS GERADOS PELO APRENDIZADO DE MÁQUINA 102
8 CIRCUITO ELETRÔNICO DE DISCRIMINAÇÃO DAS CORES RGB 103
8.1 RESULTADOS PARCIAIS DO CIRCUITO ELETRÔNICO DISCRIMINADOR 109
9 CONCLUSÃO 110
PUBLICAÇÕES GERADAS DURANTE O PERÍODO DO DOUTORADO 112
REFERÊNCIAS 113
APÊNDICE A - *PROGRAMA PARA TESTAR OS ALGORITMOS AM* 118
APÊNDICE B- *ALGORITMO LINEAR DISCRIMINANT* 119
APÊNDICE C- *ALGORITMO QUADRATIC DISCRIMINANT* 121
APÊNDICE D - *ALGORITMO LINEAR SVM* 123
APÊNDICE E - *ALGORITMO QUADRATIC SVM* 125
APÊNDICE F - *ALGORITMOS CUBIC SVM* 127
APÊNDICE G - *ALGORITMO COSINE KNN* 129
APÊNDICE H- *ALGORITMO ESNSEMBLE SUBSPACE DISCRIMINANT* 131

1 INTRODUÇÃO

O mercado de sensores de imagem tem tido um crescimento explosivo nos últimos anos, devido às crescentes demandas de imagens digitais, câmera digitais, imagens móveis, sistemas áudio/vídeo, sistemas de vigilância, dados biométricos, inspeção industrial, robótica, telecomunicações, aplicações aeroespaciais e IOT (*Internet of Things* – Internet das coisas) (GAMAL, 2005).

Os dispositivos de carga acoplada CCD (*Charged Couple Device*) tradicionalmente foram à tecnologia dominante para os sensores de imagens, prevalecendo desde a sua invenção em 1975 devido a sua faixa dinâmica superior e inferior, pixels (*Picture Element*) menores e maior sensibilidade a luz (BIGAS et al., 2006).

No início de 1990, os sensores de imagem CMOS (*Complementary Metal Oxide Semiconductor*) teve um grande avanço em relação ao seu concorrente CCD, que estavam presentes em todas a máquinas fotográficas, graças a evolução da tecnologia CMOS. As matrizes de pixels passivos CMOS foram as primeiras gerações dos sensores de imagem. Com a evolução da tecnologia e com a adição de um amplificador por coluna ou por linha, foi possível melhorar a relação sinal ruído (MADEIRA, 2012). Sendo assim, os sensores de imagem CMOS começaram a ser uma alternativa interessante na substituição da tecnologia CCD, devido à integração das funções de imagem e do processamento de sinais condensados dentro de um único chip, reduzindo o custo de fabricação e incluindo uma série de outras funcionalidades (AFZALIAN, 2006; MADEIRA, 2012)

Com o passar dos anos o consumo de tecnologias CCD/CMOS vêm crescendo de forma exorbitante, como exemplo as câmeras digitais, devido ao grande avanço da conectividade em redes sociais e "*uploads*" de imagens e vídeos. Entretanto, há muitos critérios que podem ser usados sabiamente para a escolha de um sensor de imagem. Alguns deles são subjetivos/qualitativos e outros são objetivos/quantitativos. Como exemplo, para uma câmera de celular devem ser considerados alguns detalhes como: densidade de pixels e sensibilidade, que são fatores respectivamente relacionados ao tamanho reduzido de uma câmera de celular e a qualidade na captação das imagens (F.M, 2011; GIMENES GRADOS, 2003).

Com o avanço das tecnologias o mercado atual de imagens utiliza em suma dois sensores de imagem: CCD e CMOS, que são dispositivos de estado sólido, fabricados em silício. Estes sensores são capazes de capturar imagens coloridas, utilizando um conjunto de filtros ópticos, conhecido como "*Color Filter Array*", que permitem a passagem de três comprimentos de

ondas, corresponde as cores RGB (*Red-Green-Blue*) (FARIAN; LEÑERO-BARDALLO; HÄFLIGER, 2015).

Para obter esse nível de tecnologia a comunidade científica está gerando esforços nas pesquisas dos detectores de luz, que são os elementos mais importantes na constituição dos sensores de imagem (YANG et al., 2019). Estes detectores têm como objetivo extrair uma informação óptica (luz) que corresponda a um sinal elétrico, correlacionando potência óptica, frequência e comprimento de onda, com corrente elétrica e tensão elétrica (FINDLATER et al., 2001; SZE, 1981). Sendo assim, os semicondutores são grandes candidatos para executar esse trabalho, por apresentarem propriedades ópticas e elétricas para diferentes materiais, como exemplo, silício e germânio (MENON et al., 2012). Um dos detectores de luz mais utilizado é o fotodetector PIN de silício, constituído de uma região P^+ e uma região N^+, intercalado por uma região intrínseca que pode ser do tipo N^- ou P^-. Estes dispositivos podem ser fabricados lateralmente e verticalmente (empilhados), possuindo o mesmo comportamento de tensão e corrente quando polarizados (COLINGE; COLINGE, 2002). Na grande maioria das aplicações, os dispositivos laterais são os mais utilizados devido ao custo reduzido e menor complexidade durante o processo de fabricação quando comparado aos dispositivos verticais (HAGIWARA, 2020; WONG et al., 1998).

1.1 MOTIVAÇÃO

Embora os fotodetectores PIN sejam os mais utilizados nos sensores de imagem, ainda existem muitas lacunas no que diz respeito ao seu funcionamento, no quesito discriminar cores RGB (GALAL et al., 2020; KAWADA et al., 2010; SAKAI et al., 2009). Em virtude destas lacunas, a motivação deste trabalho foi fazer os seguintes questionamentos: Será possível substituir os filtros RGB, que são filtros físicos depositados sobre o fotodetector PIN capazes de captar somente três comprimentos de onda, por um dispositivo capaz de captar outros comprimentos, além do RGB, como exemplo, o amarelo ou o violeta? Será possível desenvolver um modelo matemático, que na sua resposta pudesse informar qual a cor detectada em função do comprimento de onda e da tensão elétrica? Se os dois questionamentos anteriores pudessem ser respondidos, será possível propor uma arquitetura, composto de circuitos eletrônicos integrado com a inteligência computacional para detecção da radiação eletromagnética? Para abrir estes horizontes, a interdisciplinaridade entre as áreas da engenharia poderá dar as devidas respostas para estes questionamentos, relacionando a microeletrônica com a computação. No que diz respeito sobre a computação, a IA (Inteligência Artificial) pode

ser entendida como a capacidade de uma máquina computacional aprender e tomar decisões com o mínimo de interferência humana. Neste universo, existem campos de atuação, como exemplo, Aprendizado de Máquina (AM), Ciência de Dados e "*Big Data*", que permitirá entender e ensinar as tomadas de decisões das máquinas (HASTIE; TIBSHIRANI; FRIEDMAN, 2009; ROSS et al., 2013; VONRUEDEN et al., 2021). O aprendizado é baseado em experiências, ajustando-se as novas entradas e performando as devidas decisões para a realização de uma dada tarefa. Graças a essa inovação, computadores são capazes de: reconhecer padrões faciais, detectar fraudes bancárias, detectar e-mails com características SPAM (*Sending and Posting Advertisement in Mass*), ofertar produtos recomendadas pela Amazon ou Netflix e entre outros (LU; LI, 2020).

Com a proposta da IA, os questionamentos anteriores poderão ser respondidos, propondo nesta tese uma arquitetura capaz de discriminar as cores RGB através do fotodetector PIN lateral com porta auxiliado pelo aprendizado de máquina sem a necessidade da utilização de filtros ópticos discretos.

1.2 ORGANIZAÇÃO DA TESE

A tese será organizada em nove capítulos, sendo que no primeiro foram apresentadas uma breve introdução e a motivação deste trabalho. No segundo capítulo será feita uma revisão bibliográfica dos conceitos básicos sobre fotodetectores desenvolvidos em tecnologia CMOS.

No terceiro capítulo serão apresentados alguns sensores ópticos em tecnologia CCD e CMOS.

No quarto capítulo são apresentados os conceitos de AM, vislumbrando os métodos e algoritmos de aprendizado.

No quinto capítulo são apresentados os materiais e métodos utilizados nesse trabalho.

No sexto capítulo são apresentados os resultados e médias experimentais do fotodetector PIN lateral com porta usado para discriminar as cores RGB.

No sétimo capítulo são apresentados os resultados dos algoritmos de discriminação da radiação RGB.

No oitavo capítulo é apresentado o circuito discriminador RGB usando o fotodiodo PIN lateral com porta.

No nono capítulo são apresentados as conclusões obtidas ao longo desta tese, através de análises dos dados caracterizados auxiliados pelos algoritmos do AM e da arquitetura final proposta para discriminar as cores RGB.

2 CONCEITOS BÁSICOS SOBRE FOTODETECTORES CMOS

Os fotodetectores são dispositivos semicondutores que possuem a capacidade de converter o sinal óptico em um sinal elétrico (conversão dos fótons em pares elétron-lacuna). Podem ser usados para detectar a presença ou ausência de pequenas quantidades de luz, podendo ser calibrados para medições com intensidades abaixo de 1pW/m^2 e acima de 1mW/m^2 (AFZALIAN, 2006). Os fotodetectores em silício são utilizados em diversas aplicações, como: instrumentos ópticos, comunicações ópticas, caracterização de superfícies, sensores de imagem, análise de DNA, sensores de posição, instrumentação analítica, fotográfica, espectroscopia, entre outros.

No estudo da foto detecção, espera-se que toda a energia luminosa possa ser convertida em um sinal elétrico, garantindo que o desempenho seja de 100% para os índices de méritos utilizados para medir sua eficácia. Como isso não é possível de acontecer devido as perdas naturais ocorridas no dispositivo, a comunidade científica busca novas alternativas no quesito "tecnologia" e "topologia" para que essas perdas sejam as menores possíveis. Neste capítulo será abordado os conceitos básicos de fotogeração e absorção, tipos de fotodetectores e desempenho dos fotodetectores.

2.1 FOTOGERAÇÃO E ABSORÇÃO

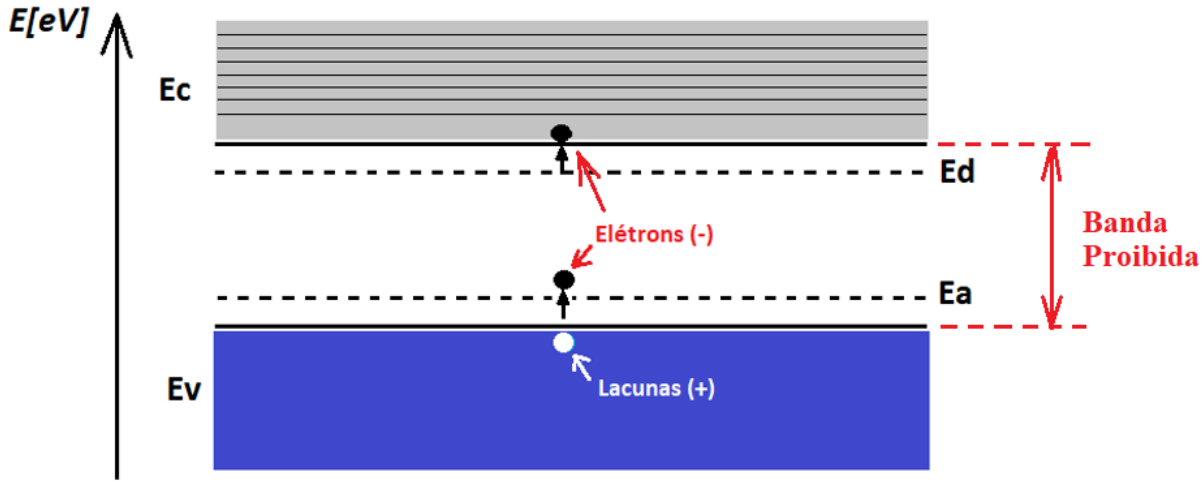
A energia luminosa (*E*) é quantizado em (*hf)*, conhecido como fóton, sendo (*f)* a frequência e (*h)* a constante de Planck de valor igual a 6,63x10^{-34}J.s = 4,14x10^{-15}eV.s, de acordo com a equação (1).

$$E = hf \tag{1}$$

Quando a luz incide sobre um material semicondutor, como exemplo o silício, a energia luminosa é transferida para os elétrons das camadas mais externas, mais precisamente da camada de valência (EV), fazendo com que o elétron ganhe energia suficiente para transitar da camada de valência para a camada de condução (EC). A partir deste ponto é gerado os pares elétron-lacuna (e-h) no semicondutor que irão contribuir para a corrente do fotodetector. A introdução de átomos doadores (arsênio ou fósforo) ou aceitadores (boro) nos semicondutores, cria-se níveis de energia dentro da faixa "proibida", com intuito de facilitar a transição dos elétrons entre os níveis de energia, conforme Figura 1. Na introdução de átomos doadores, o nível está localizado alguns meV (mili elétron-volt) abaixo da banda de condução, sendo que

em temperatura ambiente, os elétrons presentes neste nível possuem energia térmica suficiente para saltar do nível (E_d) para o nível de condução. Ao conter impurezas doadoras, o semicondutor é dito do tipo N e a concentração destes átomos é denominada de (N_D). O mesmo processo ocorre ao introduzir os átomos aceitadores, o nível de energia está localizado alguns meV acima da banda de valência, permitindo que os elétrons saltem de banda de valência para o nível (E_a), criando assim, lacunas na camada de valência. Ao conter impurezas aceitadoras o semicondutor é dito do tipo P e sua concentração é chamada de (N_A)(ABID; KHOKHAR; RAHMAN, 2011; COLINGE; COLINGE, 2002).

Figura 1 – Níveis de energia dentro da faixa proibida.



Fonte: Autor

A concentração de elétrons e lacunas para um material tipo P e N é dada pelas equações (2) e (3).

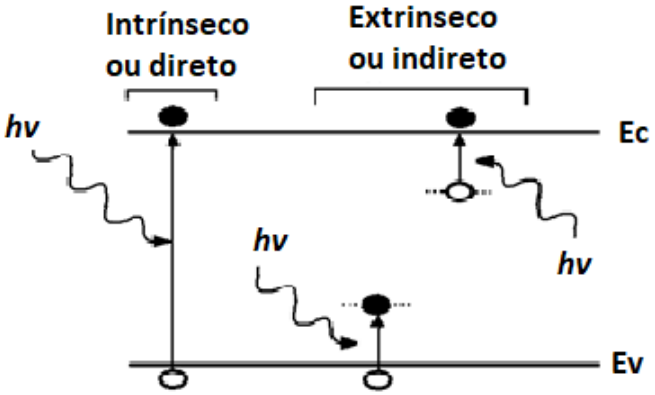
$$n_N \cong N_D \; e \;\; p_n = \frac{{n_i}^2}{N_D} \; (Semicondutor\ tipo\ N) \tag{2}$$

$$p_p \cong N_A \; e \;\; n_p = \frac{{n_i}^2}{N_A} \; (Semicondutor\ tipo\ P) \tag{3}$$

Durante a geração dos pares (e-h) nos semicondutores, os fenômenos de "geração" e "recombinação" ocorrem de forma contínua afetando diretamente o rendimento dos fotodetectores, ou seja, elétrons e lacunas sofrem um contínuo processo de geração e recombinação térmica, sendo que no equilíbrio as taxas de geração (G_{TER}) e recombinação (R_{TER}), se igualam, mantendo as concentrações dos portadores em equilíbrio, com seu produto satisfazendo a relação ($n.p = ni^2$), sendo (n e p) respectivamente as concentrações de elétrons e

lacunas num semicondutor intrínseco (JACOBUS, 2011). O processo de recombinação pode ser entendido quando um elétron da camada de condução transita para a de valência. Ao transitar, o elétron libera energia, e o mesmo se recombina com uma lacuna. Já o processo de geração ocorre quando um elétron transita da banda de valência para a de condução, criando uma lacuna na valência e um elétron livre na condução. No equilíbrio termodinâmico, os fenômenos de geração e recombinação são balanceados, tal que a concentração de elétrons e lacunas se mantém constante ao longo do tempo. No entanto, um estado de não equilíbrio pode ser atingido por meio de perturbações externas, como exemplo, uma fonte de luz. Sendo assim, existem semicondutores de banda "direta" e "indireta", também conhecido como "intrínseco" e "extrínseco", sendo que o silício é um semicondutor de banda indireta, onde a mínima energia da banda de condução e a máxima energia da banda de valência não ocorrem para o mesmo valor de momentum de portadores (*k*) (COLINGE; COLINGE, 2002). Esta característica faz com que a recombinação de banda para banda, através do qual, um elétron "salta" da banda de condução para a banda de valência se torne muito improvável de ocorrer, fazendo com que este tipo de semicondutor não emita luz. Portanto, o fenômeno de recombinação só pode ocorrer em níveis energéticos permitidos dentro da faixa proibida, chamados de centros de recombinação, os quais são introduzidos no semicondutor por átomos de impurezas ou defeitos cristalográficos, conforme Figura 2.

Figura 2 – Transição por foto excitação direto e indireto.



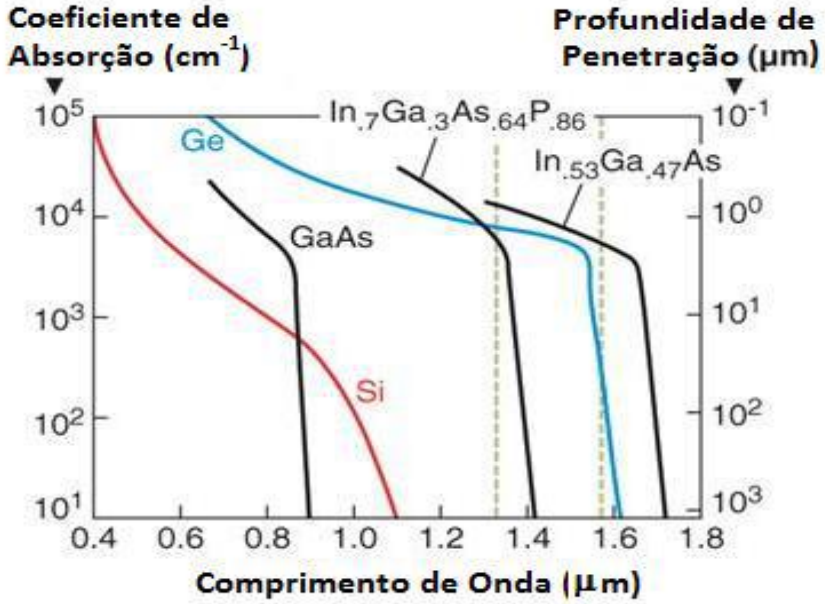
Fonte: Auto "adaptado de" NOVO, 2013

Um dos principais responsáveis pelo fenômeno de geração extrínseca nos semicondutores é a incidência da radiação luminosa, e sua absorção é determinada pelo coeficiente de absorção (α), conforme Figura 3. O silício é "transparente" a fótons com energia (*hv)* menor que a sua faixa proibida, por isso, o maior comprimento de onda (λ_C) que pode ser

absorvido pelo silício para que um elétron possa “saltar” da banda de valência para a banda de condução é dada pela equação (4).

$$\lambda_C = \frac{hc}{E} \tag{4}$$

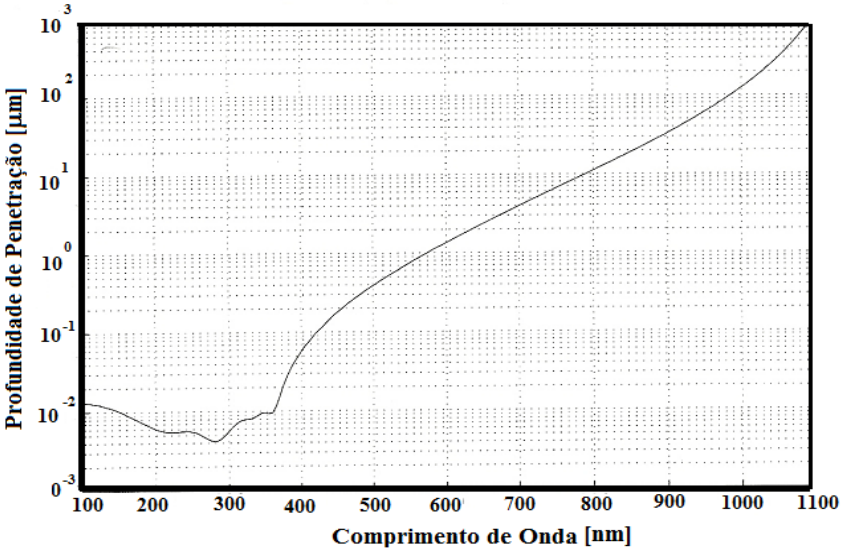
Figura 3 – Coeficiente de absorção (α).



Fonte: SZE, 1981

O inverso do coeficiente de absorção é a profundidade de penetração (L_{ABS}), conforme Figura 4, utilizando o modelo apresentado em (ZIMAN, 1972). Este coeficiente indica o quanto à luz pode penetrar no material em função do seu comprimento de onda antes de ser absorvida.

Figura 4 – Profundidade de penetração do Silício.



Fonte: AFZALIAN, 2006

A intensidade do fluxo de fótons incidente (ϕ) nos semicondutores diminui à medida que o feixe penetra no material devido ao processo de absorção visto nas Figuras 3 e 4, de acordo com a equação (5) proposto por (NEAMEN A., 2003). Sendo (ϕ_0) o valor da intensidade no fluxo de fótons incidentes na superfície e (λ) o comprimento de onda.

$$\Phi = (L_{ABS}, \lambda) = \Phi_0 . e^{-\alpha.(\lambda).L_{ABS}} \tag{5}$$

De maneira geral, fótons com energia igual ou maior que a energia da faixa proibida, podem ser absorvidos e então gerar pares (e-h). O mínimo comprimento de onda que pode ser absorvido é altamente influenciado pela recombinação de superfície, pois para baixos comprimentos de onda, o coeficiente de penetração é muito alto, portanto, a radiação é absorvida muito próxima à superfície, onde o tempo de recombinação é muito curto e os portadores fotogerados se recombinam antes de serem coletados. Através destes coeficientes é possível determinar quais fotodetectores são mais indicados para o processo fotogeração em virtude do espectro eletromagnético. Portanto, segundo (NEAMEN A., 2003), se cada fóton incidente se resulta na geração de um par (e-h), a taxa de geração desses portadores em função da profundidade de penetração é dada pela equação (6). Sendo (G_L) taxa de geração de portadores para um dado comprimento de onda e (R) a responsividade.

$$G_L(L_{ABS}, \lambda) = \alpha(\lambda). \Phi_0. [1 - R] e^{-\alpha.(\lambda).L_{ABS}} \tag{6}$$

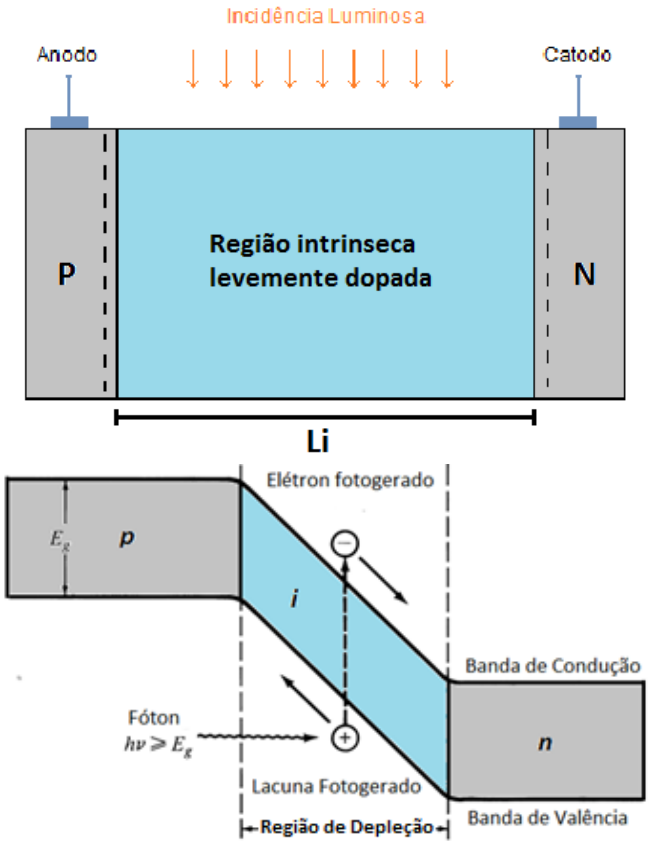
2.2 TIPOS DE FOTODETECTORES

Neste tópico serão abordados alguns tipos de fotodetectores utilizados na captação da radiação eletromagnética.

2.2.1 Fotodiodo PIN

O fotodiodo PIN (Figura 5) é mais rápido e sensível que os diodos de junção PN. Sua estrutura consiste das junções PN separadas por uma região intrínseca com dado comprimento (L_i). A região intrínseca pode ser tanto o material N ou P com baixas concentrações de portadores, conforme Figura 5. Esta região (intrínseca) pode ser totalmente depletada ou não, dependendo da tensão aplicada e da sua dopagem. Quando a luz é incidida sobre a área fotossensível (região intrínseca) são gerados pares (e-h) dentro da região de depleção (RDD), que são separados pela ação do campo elétrico, fazendo com que a corrente comece a fluir pelo dispositivo.

Figura 5 – Fotodiodo PIN.



Fonte: AFZALIAN, 2006

Além disso, dentro da RDD a taxa de recombinação é menor que nas regiões N e P, portanto, para se ter um bom rendimento, os fotodiodos PIN operam com polarizações reversas para aumentar a camada de depleção no intuito de absorver uma quantidade maior de luz. Os pares (e-h) gerados devem ser separados rapidamente pela ação do campo elétrico no intuito de evitar a recombinação. Entretanto, para se ter um campo elétrico interno elevado, é necessária uma região de depleção pequena (MARTINO; PAVANELLO; VERDONCK, 2003). Desta forma, há um compromisso na determinação da RDD, pois a mesma deve ser grande o suficiente para permitir uma maior conversão dos fótons, e suficientemente pequena para diminuir o tempo de trânsito dos portadores fotogerados (STREEMAN; BANERJEE, 2000).

Além da corrente fotogerada (I_{FG}), outro parâmetro importante no desempenho dos fotodiodos PIN, é a sua corrente de escuro (I_{ESCURO}), que nada mais é que a corrente de fuga do dispositivo quando o mesmo está polarizado reversamente, sem a incidência de luz, conforme equação (7) (NEAMEN A., 2003). Esta corrente surge devido à geração térmica dos pares (e-h) dentro da RDD e da polarização reversa, pois acima de T ≈ 0K, os elétrons podem adquirir energia suficiente para transitar da banda de valência para condução devido às vibrações da rede cristalina (COLINGE; COLINGE, 2002). Sendo (J_{ESCURO}) a densidade de corrente no escuro, (D_P) e (D_N) são respectivamente as constantes de difusão das lacunas e dos elétrons, (pn_0) é a concentração de portadores minoritários da lacunas no lado N em equilíbrio térmico, (np_0) é a concentração dos portadores minoritários dos elétrons no lado P em equilíbrio térmico, (τ_0) é definido como o tempo de vida médio entre o tempo de vida das lacunas e dos elétrons livres, (L_N) e (L_P) são respectivamente os comprimentos de difusão dos elétrons e das lacunas, que representam o quanto estes portadores conseguem percorrer desde o momento de sua geração até o momento da recombinação, (q) é a carga elementar do elétron em módulo igual a $1{,}6 \times 10^{-19}$C e (W) representa a largura da região de depleção.

$$J_{ESCURO} = \left[\frac{q.D_N.n_{p0}}{L_N} + \frac{q.D_P.p_{no}}{L_P}\right] + \frac{q.n_i.WD_P.p_{no}}{2.\tau_0} \tag{7}$$

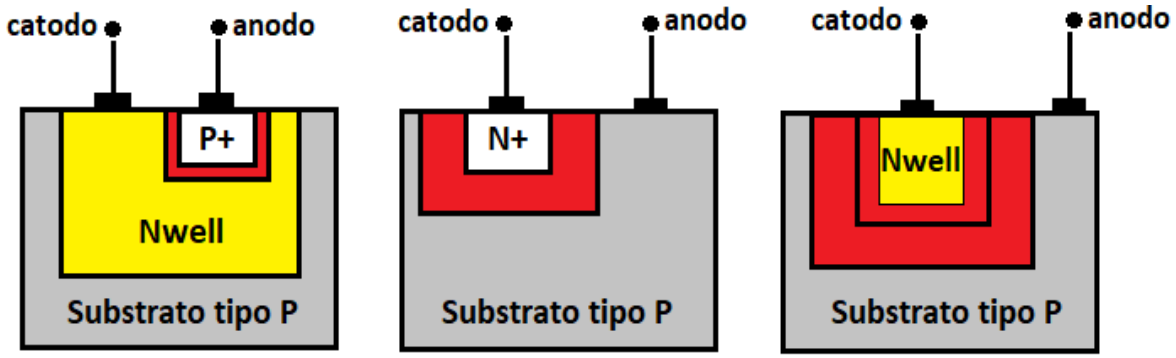
Portanto a corrente total (I_T) que flui nos fotodiodos PIN, consiste na somatória da corrente fotogerada (I_{FG}) com a de escuro (I_{ESCURO}), conforme equação (8):

$$I_T = I_{FG} + I_{ESCURO} \tag{8}$$

2.2.2 Tipos básicos de fotodiodos convencionais verticais

Os três tipos de fotodiodos convencionais que utilizam o processo CMOS padrão, são conhecidos como: $P+/N_{well}$, $N+/P_{SUB}$ e N_{well}/P_{SUB} (CAMPOS, 2008), conforme Figura 6:

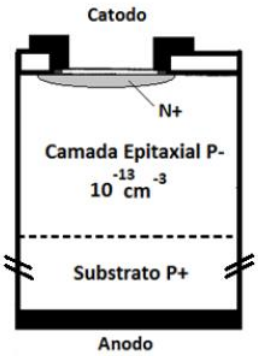
Figura 6 - Tipos básicos de fotodiodos em tecnologia CMOS: P+/Nwell, N+/PSUB e Nwell/PSUB.



Fonte: Autor

Verifica-se que a RDD formada entre as regiões (áreas em vermelho) fica em sua grande parte distante da superfície, acarretando baixa eficiência para comprimentos de onda curtos. Outra coisa importante de se observar, é que a espessura da camada de depleção poderá ter variações, pois seu controle se torna difícil, sendo que ela depende das concentrações de dopantes que é um parâmetro da tecnologia empregada. Com poucas modificações no processo de fabricação, foram desenvolvidos os fotodiodos PIN verticais, conforme Figura 7, sendo que o fotodiodo é formado entre a camada N^+, a camada epitaxial P^- e o substrato P^+. Usualmente a concentração de dopantes da camada epitaxial de $10^{15}cm^{-3}$, que consequentemente a pequena tensão reversa disponível nos circuitos integrados resultará em uma RDD de apenas 2µm, na qual, não será o suficiente para evitar o mecanismo de difusão de portadores para altos comprimentos de onda, visto que apenas parte da camada epitaxial é depletada (KIM et al., 2010; MUTTAQIN; WIJAYA; PRAJITNO, 2018; ZIMMERMANN, 2000).

Figura 7 – Fotodiodo PIN vertical.

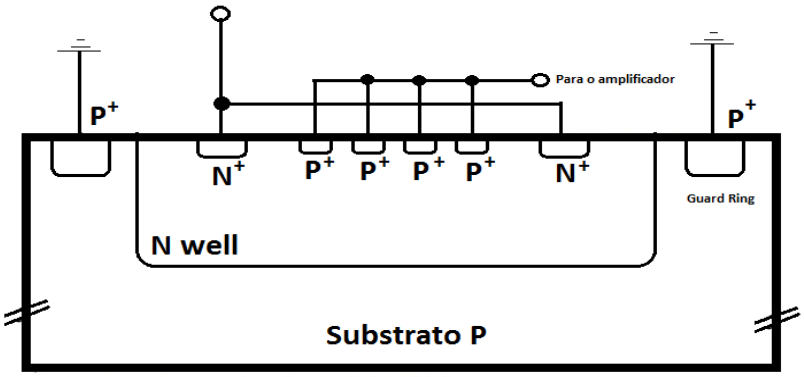


Fonte: ZIMMERMANN, 2000

2.2.3 Fotodiodos PIN Multi-Dedos

Os fotodiodos multidedos surgiram da extensão dos diodos verticais com pequenas modificações, onde a camada (P^+) foi substituída pelos dedos (P^-), a fim de melhorar a eficiência quântica para baixos comprimentos de onda. Variando a distância entre os dedos, bem como seu comprimento, pode-se melhorar a sensibilidade e velocidade de resposta (ZIMMERMANN, 2000), conforme Figura 8. Para curtas distâncias entre os dedos, os tempos de subida e descida do sinal ficaram abaixo de 0,25ns para comprimentos de onda na faixa do azul (λ=400nm), caracterizando uma largura de banda de 1GHz (NEAMEN A., 2003). Já para a faixa do vermelho com comprimentos de onda em torno de λ=750nm, os tempos de subida e descida ficaram próximos dos 30ns, devido ao retardo da difusão dos portadores fotogerados no substrato (BAKER, 1997; ELANI, 2010).

Figura 8 – Fotodiodo multi-dedo em Nwell em tecnologia CMOS convencional de 0,35µm.



Fonte: AFZALIAN, 2006

2.3 DESEMPENHO DOS FOTODETECTORES

O ideal para um fotodetector seria detectar todos os fótons incidentes independente do seu comprimento de onda e transformá-los em um sinal elétrico sem a presença de ruídos ou perdas. Entretanto, isso não é possível, pois todos os fotodetectores possuem perdas durante a conversão. Para medir a eficiência dos dispositivos será discutido algumas figuras de mérito responsáveis por estas medições.

2.3.1 Responsividade

A responsividade (*R*) é um dos principais parâmetros utilizados na caracterização dos fotodetectores, definida como sendo a relação da corrente fotogerada (I_{FG}) e a potência óptica incidente (P_{IN}) (COLINGE; COLINGE, 2002), dada pela equação (9).

$$R = \frac{I_{FG}}{P_{IN}} \tag{9}$$

Sabendo que a corrente fotogerada (I_{FG}) é a corrente total menos a corrente de escuro ($I_{TOTAL} - I_{ESCURO}$), então a responsividade pode ser calculada de outra forma, de acordo com a equação (10) (SZE, 1981).

$$R = \frac{I_T - I_{ESCURO}}{P_{IN}} \tag{10}$$

2.3.2 Relação sinal – ruído

A corrente de escuro presente em um fotodetector é na verdade, uma forma de ruído inerente a estes tipos de dispositivos. Portanto, pode ser definida uma relação sinal ruído (SNR) através da equação (11).

$$SNR = \frac{I_{FG}}{I_{ESCURO}} = \frac{I_T - I_{ESCURO}}{I_{ESCURO}} \tag{11}$$

Esta relação pode também ser expressa em decibéis (dB) sendo altamente influenciada pela temperatura (COLINGE; COLINGE, 2002).

2.3.3 Eficiência quântica

A eficiência quântica total (EQ_T) é definida como sendo a razão entre a corrente fotogerada (I_{FG}) e a máxima corrente que poderia ter sido gerada se todos os fótons fossem convertidos em portadores livres ou a energia de transferência do fóton (E) multiplicado pela responsividade (R), dada pela equação (12) (SZE, 1981).

$$EQ_T = \frac{I_{FG}}{I_{FG(máx)}} = E.R \tag{12}$$

Sabendo que energia $E = hc / \lambda$ e a responsividade $R = (I_{TOTAL} - I_{ESCURO}) / P_{IN}$, a eficiência quântica total pode ser calculada de outra maneira, de acordo com a equação (13).

$$EQ_T = \left(\frac{h.c}{\lambda}\right).\left(\frac{I_{TOTAL} - I_{ESCURO}}{P_{IN}}\right) \tag{13}$$

3 SENSORES ÓPTICOS EM TECNOLOGIA CCD E CMOS

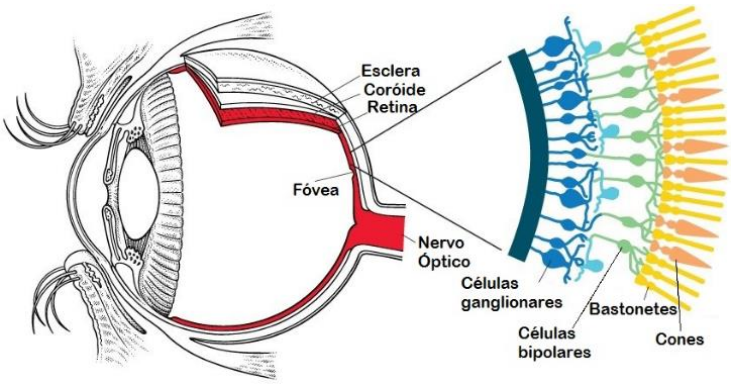
Neste capítulo serão apresentados alguns sensores ópticos utilizados para discriminar as cores do espectro eletromagnético visível dispensando a utilização dos filtros discretos. Inicialmente será apresentado um breve resumo dos estudos voltados à "Colorimetria" que é a base para um bom entendimento dos sensores bio-inspirados (LEÑERO-BARDALLO; BRYN; HÄFLIGERR, 2014) com intuito de imitar o olho humano.

3.1 COLORIMETRIA

O sistema visual humano é composto basicamente de uma rede de sensores especializados na captação e distinção das cores oriundas da luz, que compõe o espectro eletromagnético [(BRIGGS, 2007)].

A retina é a parte do olho sensível à luz e sua superfície é composta por fotorreceptores e terminações nervosas. Existem dois tipos de fotorreceptores denominados de células sensíveis à luz, chamados de "cones" e "bastonetes", os quais podem ser visualizados, conforme Figura 9.

Figura 9 – Representação do olho humano.

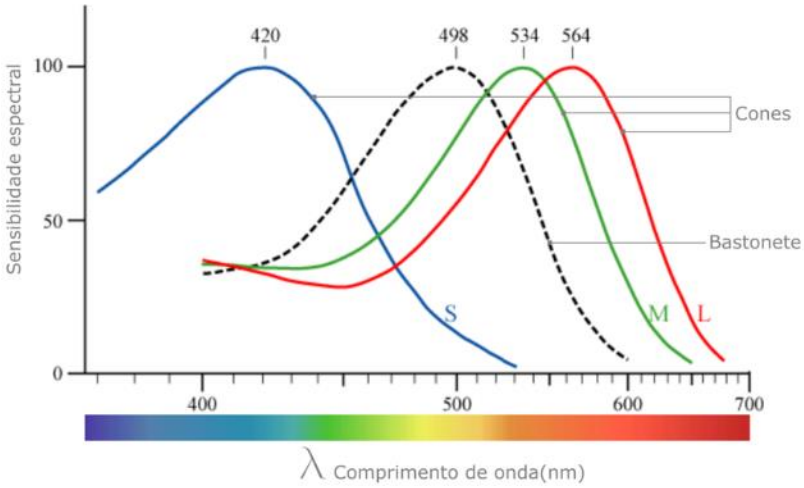


Fonte: Aplicações Informáticas B, 2015

Os cones são responsáveis por diferenciar as cores devido a sua sensibilidade para os diferentes comprimentos de onda, enquanto os bastonetes são responsáveis pela intensidade luminosa, ou seja, o quão forte a luz está para um dado comprimento de onda (WYSZECKI, 1982).

Este complexo sistema de discriminação das cores composto do olho humano juntamente com o cérebro utilizam o método mais eficiente para o processamento de grandes quantidades de comprimentos de onda. Este sistema divide o espectro visível dentro das regiões mais dominantes, vermelho representando o cone L (comprimentos longos), verde representado o cone M (comprimentos médios) e azul representando o cone S (comprimentos curtos), onde se concentram as informações das cores (MOGHAVVEMI et al., 2012; SEAUENCE et al., 1995), conforme Figura 10.

Figura 10 – Sensibilidade dos três tipos de cones da retina.



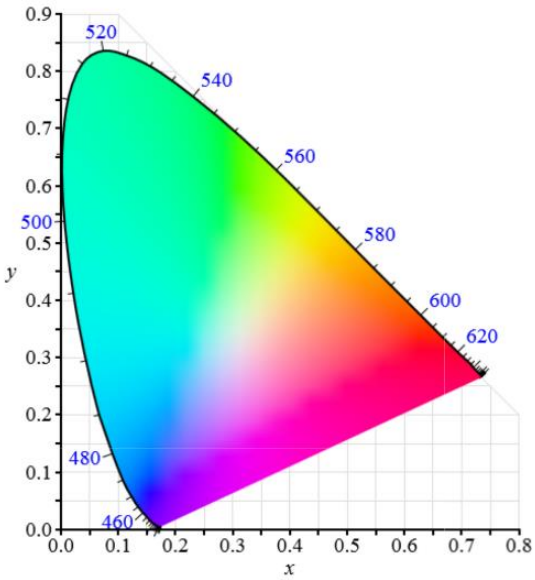
Fonte: Cor, 2010

3.1.1 Diagrama cromático CIE

O diagrama de cromaticidade do sistema CIE (*Commission Internationale de I'Eclairage* – Comissão Internacional de Iluminação), ilustrado na Figura 11, mostra os limites de todas as cores visíveis que representam as combinações de cores monocromáticos de espectro.

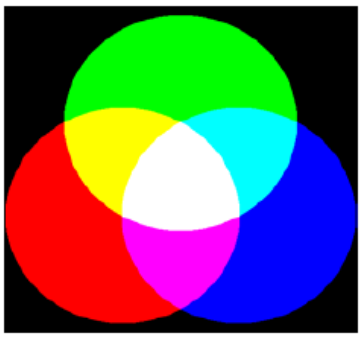
A medição das cores do sistema CIE é baseada na possibilidade de matizar qualquer cor com uma combinação de três cores primárias. A mistura aditiva, tem como cores primárias o vermelho (R), verde (G) e o azul (B), sendo que a soma destes três elementos resulta na cor branco, sendo preto a ausência total da luz, conforme Figura 12.

Figura 11 – Diagrama cromático do espaço de cores no CIE 1931.



Fonte: CIE 1931 Color Space, 2017

Figura 12 – Representação do sistema aditivo de cores.

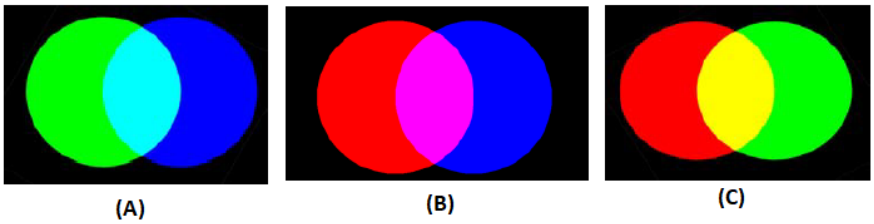


Fonte: CMYK e RGB," 2010

Quando há sobreposição de duas cores a nova cor tem nível de cromaticidade (tonalidade e croma) correspondente à média das duas cores originais e luminosidade igual à soma das duas. Logo, o resultante da combinação das cores RGB são:

a) verde com azul resulta em azul ciano (Figura 13 – A);
b) vermelho com azul resulta em magenta (Figura 13 – B);
c) vermelho com verde resulta em amarelo (Figura 13 – C);

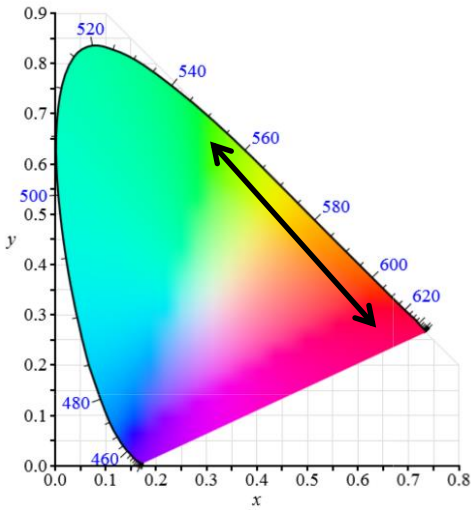
Figura 13 – Representação do sistema aditivo de cores.



Fonte: CMYK e RGB, 2010

Como exemplo, é possível verificar na Figura 13 – C, que a sobreposição da luz vermelha com verde resulta em uma luz amarela. Na Figura 14 a reta com extremidades entre a luz verde e a luz vermelha representa todas as possíveis combinações destas duas cores. Qualquer nova cor sobre essa reta teria como nível de luminosidade a soma das luminosidades das cores dos extremos.

Figura 14 – Representação do sistema aditivo de cores.



Fonte: Autor “adaptado de” CIE 1931 Color Space, 2017

3.1.2 Modelo de cores RGB e filtros CFA

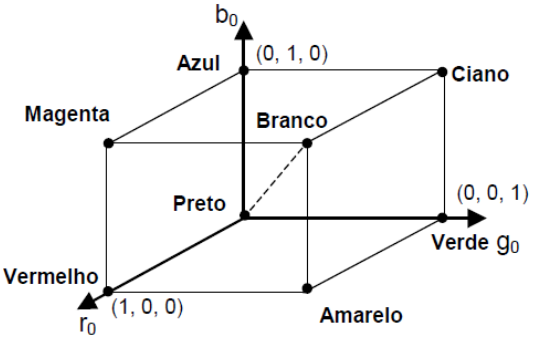
O modelo de cores RGB é um modelo aditivo que possui três cores (vermelho, verde e azul) como cores primárias (Figura 12). Através da combinação de diferentes intensidades destas três cores, é possível reproduzir uma grande variedade de cores. Câmeras de vídeo, scanners, vídeo games, e câmeras digitais são exemplos de dispositivos que utilizam este sistema para detecção. Televisores (CRT, LCD, plasma, OLED, Quantum-Dots etc.), displays de computadores e celulares, vídeo projetores, display de LEDs e grandes telas, como a *JumboTron*, são exemplos de dispositivos que utilizam este sistema para reprodução de cores (BRIGGS, 2007).

O primeiro experimento com o modelo RGB em fotografia foi realizado em 1861 por James Clerk Maxwell, envolvendo três diferentes "*takes*" de cores filtradas (HIRSCH, 2004). Para realizar a fotografia, três projeções alinhadas sobre uma tela em uma sala escura foram necessárias.

Cada componente de cor representa um eixo ortogonal em um espaço euclidiano (Figura 14) normalizado definido conforme equação (14) (WYSZECKI, 1982).

$$r = \frac{R}{R_{máx}}, \qquad g = \frac{G}{G_{máx}}, \qquad b = \frac{B}{B_{máx}} \tag{14}$$

Figura 15 – Cubo do modelo de cores RGB.



Fonte: WYSZECKI, 1982

As componentes de cores Rmáx, Gmáx e Bmáx são as intensidades máximas de cores para cada componente de cor correspondente e *r, g* e *b* são os componentes RGB normalizados em valores de 0 a 1. Esse modelo, trata uma imagem colorida como um conjunto de três imagens

independentes em tons de cinza, cada uma das quais representando as cores, vermelho, verde e azul.

Seis dos oito cantos do cubo (Figura 15) descrevem as três cores primárias vermelho, verde e azul e as três cores secundárias amarelo, magenta e ciano. Os dois cantos adicionais correspondem a todas as combinações de valores iguais dos três componentes de cor, o que resulta nos possíveis níveis de cinza que uma imagem digital pode conter neste padrão [(HIRSCH, 2004; WYSZECKI, 1982)].

A variação dos valores pode ser quantificada de diversas formas:

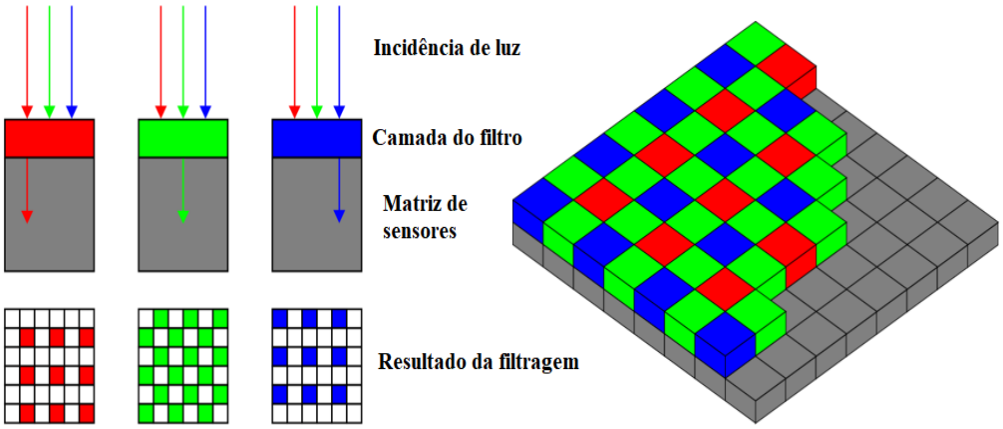
a) De 0 a 1, com qualquer fração intermediária. Sendo este utilizado em análises teóricas e sistemas com representação de ***ponto flutuante*** em computação;
b) Cada valor pode também ser uma porcentagem, de 0% a 100%;
c) Em computação, é comum utilizar um valor inteiro para cada componente variando entre 0 e 255 (variação que um único byte consegue representar);
d) Equipamentos de imagens digitais de alta fidelidade (os chamados *high-end*) permitem maiores variações de valores, tais como de 0 a 1023 (10 bits), de 0 a 65535 (16-bits) e até 32-bits, 48-bits e 64-bits;

Em televisores coloridos e câmeras de vídeos fabricados antes dos anos 90, a luz incidente era separada por prismas e filtros em três cores primárias (tons de R, G e B) e essas três componentes incidiam em três tubos de raios catódicos. Com o surgimento da tecnologia CCD, nos anos 1980, e com integração de circuitos em alta escala, os tubos catódicos foram substituídos por inviabilidade industrial. Os dispositivos compostos por sensores foram simplificados, tornando-os menores, com menos componentes ópticos e mais baratos. Os fotossensores utilizados tinham a capacidade de detectar as intensidades de luz, mas não o seu comprimento de onda, sendo assim, os filtros discretos foram implantados no intuito de detectar pelo menos as três cores primarias (RGB) para compor uma imagem colorida (NAKAMURA, 2005).

Um filtro CFA (*Color Filter Array*), ou CFM (*Color Filter Mosaic*), é um mosaico, formado por pequenos filtros de cor, colocado sobre um sensor de luz para captar informação da cor. A finalidade dos filtros é deixar passar somente o comprimento de onda corresponde a sua cor. Por exemplo, o filtro Bayer (Figura 16), amplamente utilizado na indústria, fornece ao sensor informações sobre intensidade do vermelho, verde e azul. Como visto na Figura 10, o ser humano é mais sensível aos comprimentos de ondas entre o verde e vermelho dentro do

espectro visível, por este motivo existem mais pontos verdes no filtro de Bayer do que vermelhos e azuis (proporção de 1:2:1 para vermelho, verde e azul respectivamente), pois desta forma as imagens captadas parecem mais nítidas para um observador humano (HIRSCH, 2004).

Figura 16 – Filtro discreto CFA ou CFM.



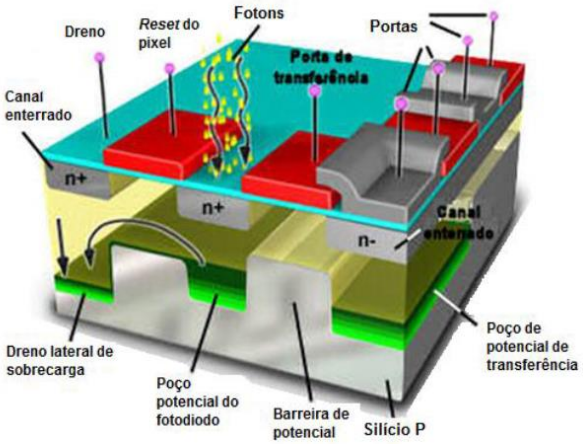
Fonte: Autor “adaptado de” Filtro Bayer, 2017

3.2 SENSOR DE IMAGEM CCD

Tal como os microprocessadores e os circuitos integrados de memória, o CCD (dispositivo de carga acoplada) é feito em lâminas de silício utilizando o processo de fotolitografia para definir e construir elementos funcionais diferentes dentro do microcircuito.

Os CCDs baseiam-se em uma densa matriz de fotodiodos, os quais operam na conversão de energia luminosa em carga elétrica. São fabricados utilizando tecnologia MOS (*Metal-Oxide Semiconductor*) específica, onde cada pixel pode ser considerado como um capacitor MOS, conforme Figura 17, que converte fótons em carga elétrica que é armazenada para uma leitura posterior (CAMPOS, 2008).

Figura 17 – Sensor de imagem CCD.



Fonte: LIMA, 2006

As arquiteturas dos dispositivos CCDs geralmente são classificadas em função do mecanismo de transferência de quadros (*frames*) e pertencem a uma das quatro categorias: transferência de quadro completo (*Full Frame*), transferência de quadro (*Frame Transfer*), transferência parcial de quadro (*Split Frame Transfer*) e transferência entre linhas (*Interline Transfer*) (CAMPOS, 2008).

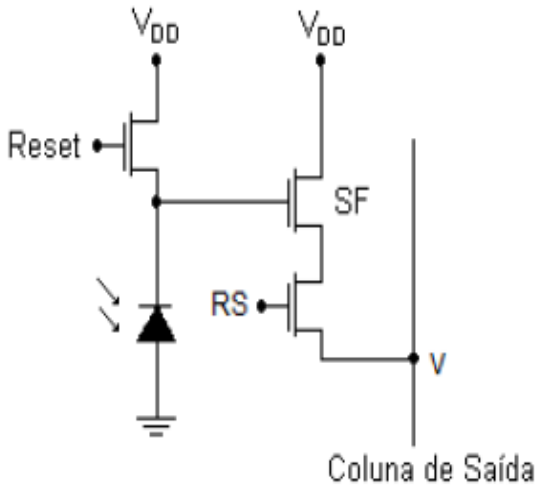
3.3 SENSOR DE IMAGEM CMOS

Assim como os CCDs, os sensores de imagem CMOS, são compostos por elementos sensíveis à luz e em função da intensidade luminosa incidente sobre eles, geram um sinal elétrico ou carga elétrica. Em cada tecnologia, o processo utilizado para realizar esta conversão é diferente. No CCD o pixel utiliza um capacitor MOS enquanto nos sensores CMOS, o pixel pode ser constituído de fotodiodos, capacitores e transistores que podem ser integrados formando um circuito individual para cada pixel (MADEIRA, 2012). Existem basicamente dois tipos de sensores de imagem CMOS. O pixel passivo, no qual existem somente componentes passivos, e o pixel ativo, que possui circuitos de amplificação. Ambos utilizando as mesmas técnicas para leitura dos pixels, nos dois casos existem a possibilidade de selecionar as linhas e colunas permitindo acessar pixels individuais ou somente a região de interesse dentro da imagem. Assim, é possível aumentar o número de quadros lidos por segundo (CHMIELEWSKI, 2009).

3.3.1 Sensor CMOS de pixel ativo (APS)

A célula básica de um fotodiodo APS (LIMA, 2006) é composta por um fotodiodo e três transistores, sendo: um transistor de *Reset*, um transistor de Seleção de Linha (RS) e um transistor Seguidor de Fonte (SF), conforme Figura 18.

Figura 18 – Sensor de imagem de pixel ativo.

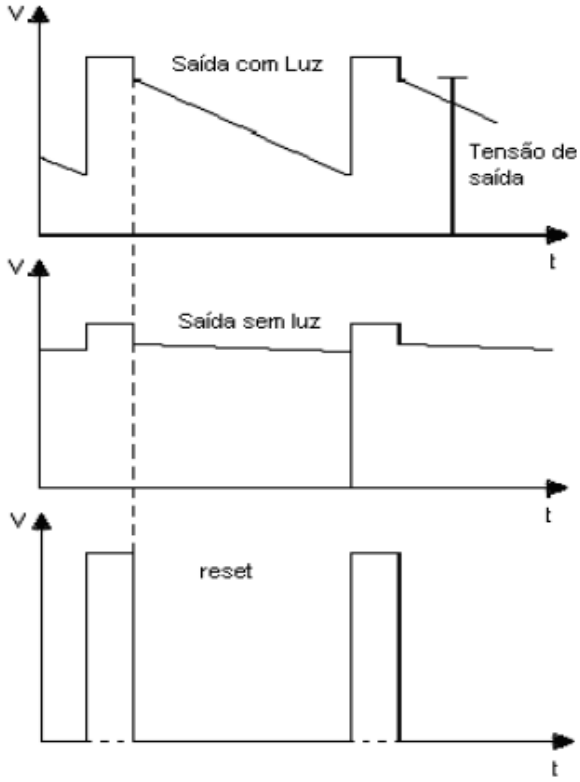


Fonte: Filtro Bayer, 2017

O transistor *Reset* é usado para inicializar o pixel, carregando a capacitância de junção do fotodiodo. Esta carga gera uma tensão V_{DD}-V_T (V_T corresponde a tensão de limiar do transistor e V_{DD} é a tensão de alimentacão) na coluna de saída que cai em função do nível de energia luminosa. Consequentemente a tensão entre porta e fonte do transistor SF também decai, diminuindo a largura de região de depleção do fotodiodo. O transistor RS seleciona o pixel da matriz cujo sinal é conduzido para o transistor de carga.

Os gráficos apresentados na Figura 19, representam o funcionamento da célula APS. Quando o transitor "*Reset"* é acionado, a capacitância de junção do fotodiodo é carregada com um certo valor de tensão. O próximo estágio é o tempo de integração, que começa quando o transistor de "*reset"* é comutado para o estado desligado. Durante este período de integração, a fotocorrente descarrega a capacitância do fotodido. Após o processo de integração, finalmente o valor do sinal de saída é lido através do seguidor de fonte e do transistor de seleção.

Figura 19 – Sinal de saída do pixel ativo com e sem incidência de luminosidade.



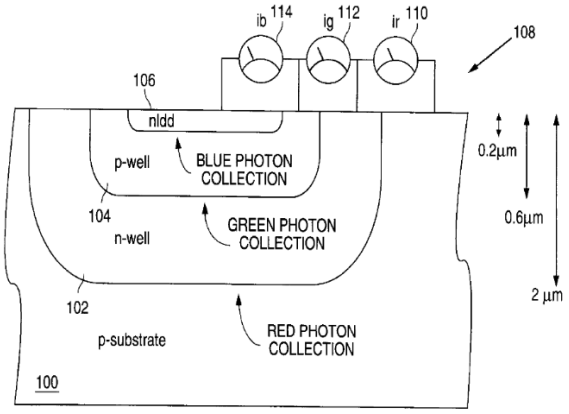
Fonte: MADEIRA, 2012

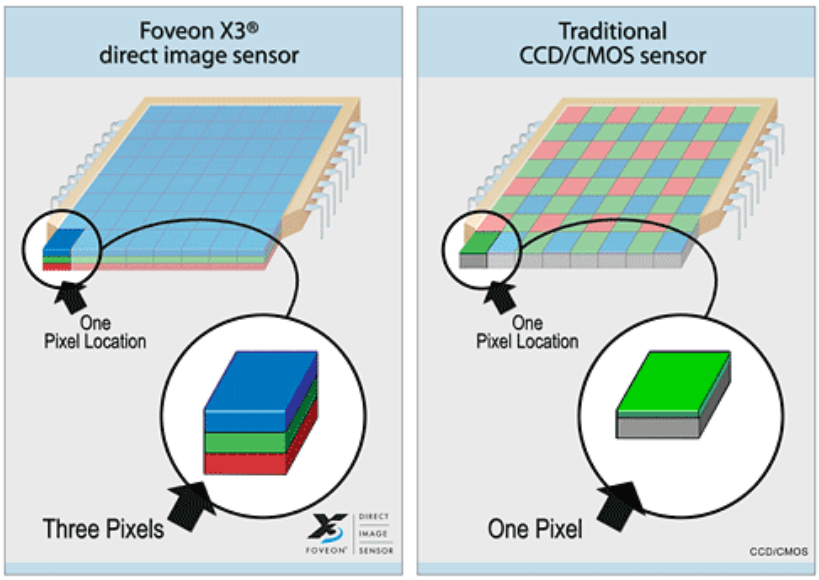
3.4 SENSOR ÓPTICO FOVEON X3

Dispositivo desenvolvido por Richard Billings Merrill em 1998, chamado *Foveon X3*, utiliza a diferença da profundidade de absorção de cada comprimento de onda como ferramenta para discriminação de cores, conforme Figura 20. Para a detecção das cores primárias RGB, o sensor óptico dispensa à utilização de filtros discretos CFA, conforme visto nos capítulos anteriores (FOVEON; MERRILL, 1999).

Este dispositivo pode ser caracterizado eletricamente como uma pilha (vertical) de fotodiodos de junções n-p-n-p, onde os comprimentos de onda correspondente à cor azul são captados na junção mais rasa, aqueles correspondentes à cor verde são captados na junção intermediária e o correspondente à cor vermelha são captados na junção mais profunda do material. Uma das caraterísticas mais importantes deste dispositivo é o tamanho da área fotossensível não ser muito menor que o espaçamento entre os sensores ao redor, como ocorre em sensores que utilizam filtros CFA, além de serem fabricados pelo processo CMOS convencional (FOVEON; MERRILL, 1999).

Figura 20 – Sensor *Foveon X3*.



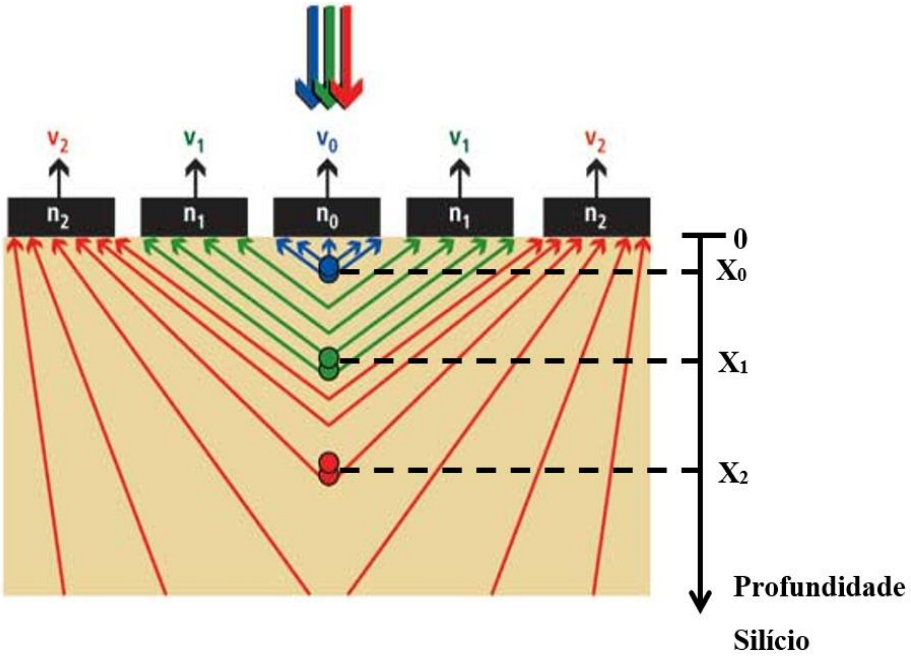


Fonte: FOVEON; MERRILL, 1999

3.5 SENSOR ÓPTICO TFD

O dispositivo TFD (*Transverse Field Detectors*) possui a mesma finalidade do sensor *Foveon X3* em discriminar diferentes cores sem a utilização de filtros discretos CFA. Assim como no *Foveon X3*, o TFD utiliza a propriedade da profundidade de penetração dos comprimentos de onda para a diferenciação das cores. Entretanto, sua estrutura não possui junções empilhadas na vertical onde estão localizadas as profundidades correspondentes às regiões de absorção dos fótons, mas sim, a utilização de campos elétricos para fazer o mesmo trabalho que as junções fariam (LONGONI et al., 2008), de acordo com a Figura 21.

Figura 21 – Sensor TFD.



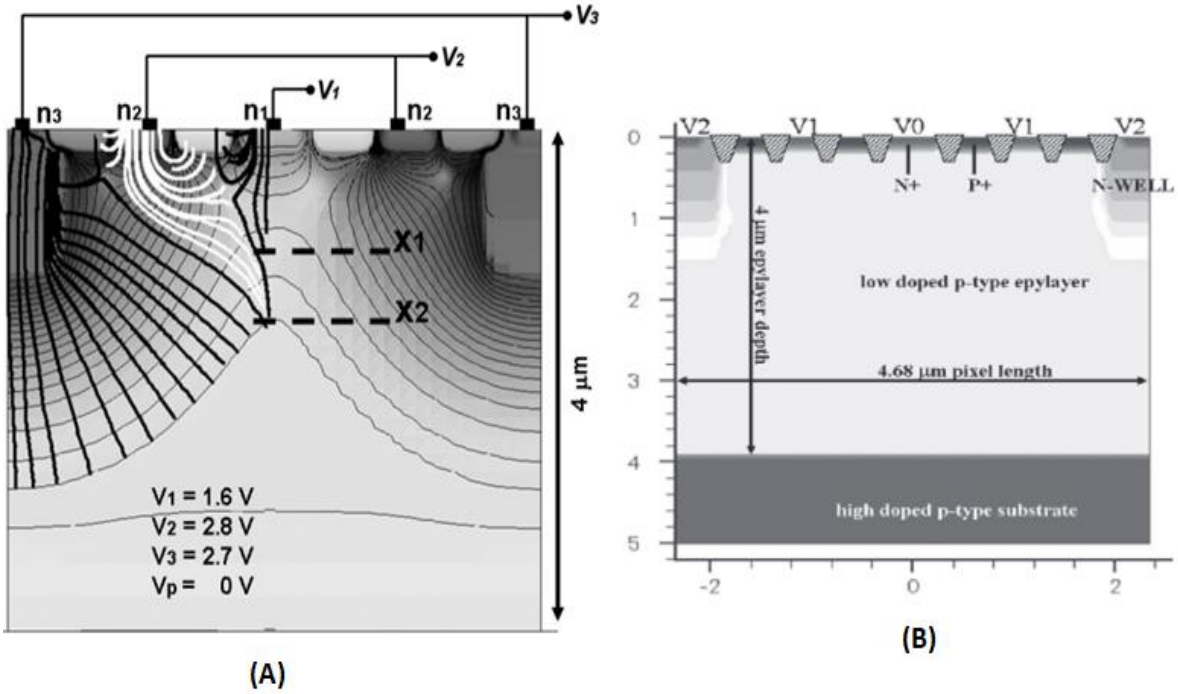
Fonte: Autor "adaptado de" Novel Sensors Help Smart Cameras Serve Niches, 2012

Sabe-se que baixos comprimentos de onda possuem profundidades de penetração de algumas centenas de nanômetros, enquanto os altos de comprimentos de onda são absorvidos em profundidades de 4 a 5µm no substrato (LONGONI et al., 2008). Sendo assim, este dispositivo baseia-se na aplicação de diferentes tensões em seus eletrodos para a captação dos fótons em diferentes profundidades, em virtude dos campos elétricos gerados, conforme Figura 22 – A.

A estrutura do dispositivo consiste em regiões intercaladas N^+ / P^+ entre os eletrodos, regiões do tipo N em suas extremidades, camada do material tipo P fracamente dopada com espessura de 4µm depositada sobre o substrato tipo P altamente dopado, conforme Figura 22 – B.

As linhas de campo elétrico gerado pelo eletrodo n_1 capta os portadores fotogerados para comprimentos de ondas que atinjam a profundidade X_1, o eletrodo n_2 para a profundidade X_2 e o eletrodo n_3 capta o restante dos portadores da região depletada e da camada epitaxial (LONGONI et al., 2008).

Figura 22 – A) Linhas de campo elétrico do sensor TFD, B) Estrutura do sensor TFD.

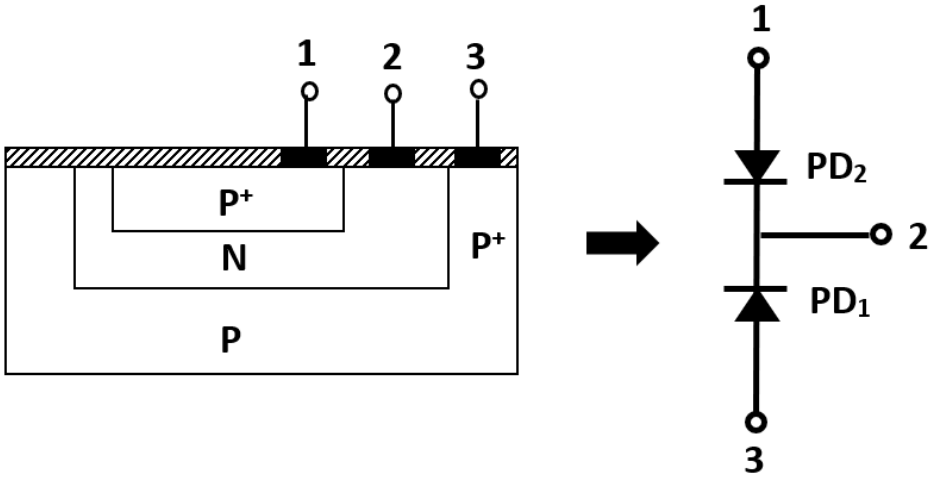


Fonte: LONGONI et al., 2008

3.6 SISTEMA DE DETECÇÃO DE COMPRIMENTO DE ONDA E INTENSIDADE LUMINOSA

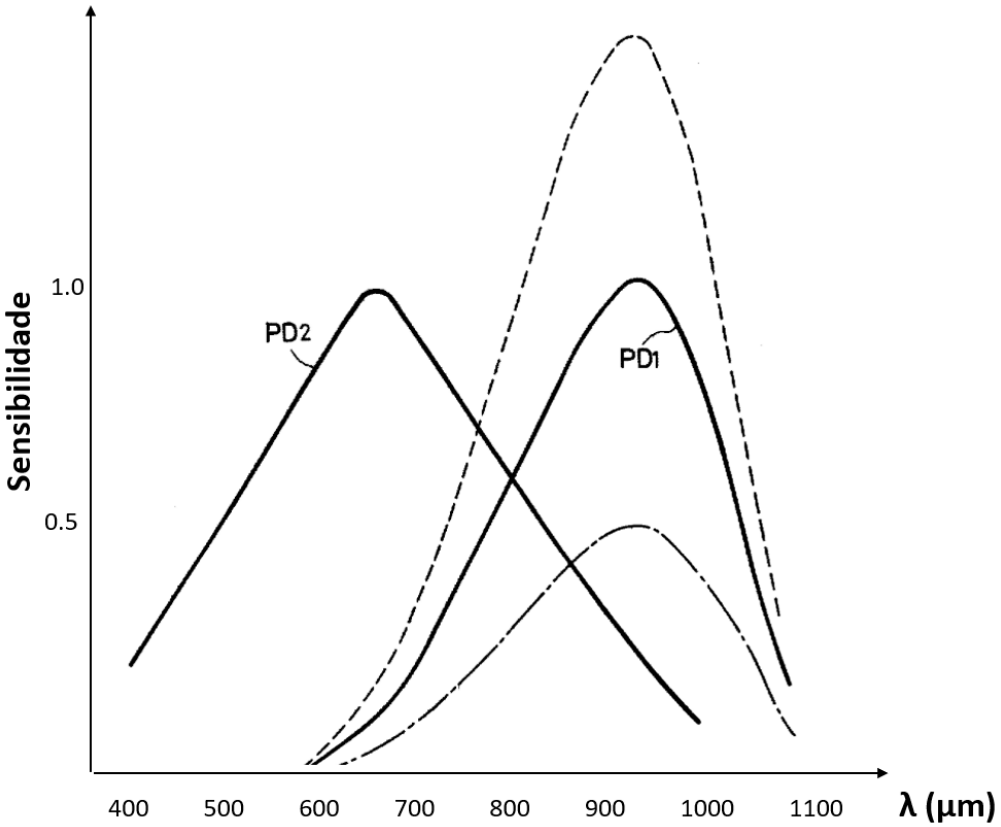
O sistema de detecção de comprimento de onda e intensidade luminosa foi criado e patenteado em 1982 por (ASO; KAWANABE, 1982) composto de circuitos eletrônicos e um fotodetector vertical com perfil $P^{+}N$ P, apresentado na Figura 23. O fotodetector é composto de duas regiões PN, sendo PD_1 a região para detectar comprimentos de ondas longos e PD_2 para detectar comprimentos de ondas curtos, conforme responsividade extraída na Figura 24, abrangendo um range de captação dos comprimentos de onda entre 400 µm à 1100 µm.

Figura 23 – Fotodetector vertical com perfil P^+-N-N proposto por (ASO; KAWANABE, 1982).



Fonte: Autor “adaptado de” ASO; KAWANABE, 1982

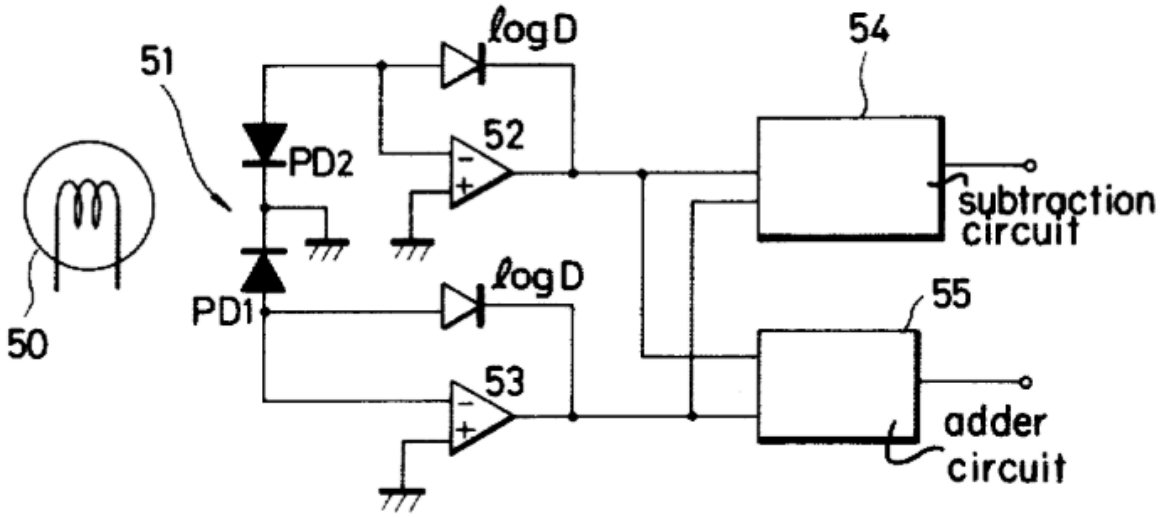
Figura 24 – Responsividade do fotodetector vertical proposto por (ASO; KAWANABE, 1982)



Fonte: ASO; KAWANABE, 1982

O sistema de detecção é composto basicamente de dois amplificadores operacionais logarítmicos, um circuito somador e um circuito subtrator, conforme Figura 25. O sistema funciona da seguindo maneira: quando a luz é submetida sobre o fotodetector (lâmpada - 50), a junção PD_1 detecta os comprimentos longos e a junção PD_2 detecta os comprimentos curtos, que serão coletados e amplificados pelos respetivos amplificadores logarítmicos (53) e (52).

Figura 25 – Sistema de detecção de comprimento de onda e intensidade luminosa proposto por (ASO; KAWANABE, 1982).



Fonte: ASO; KAWANABE, 1982

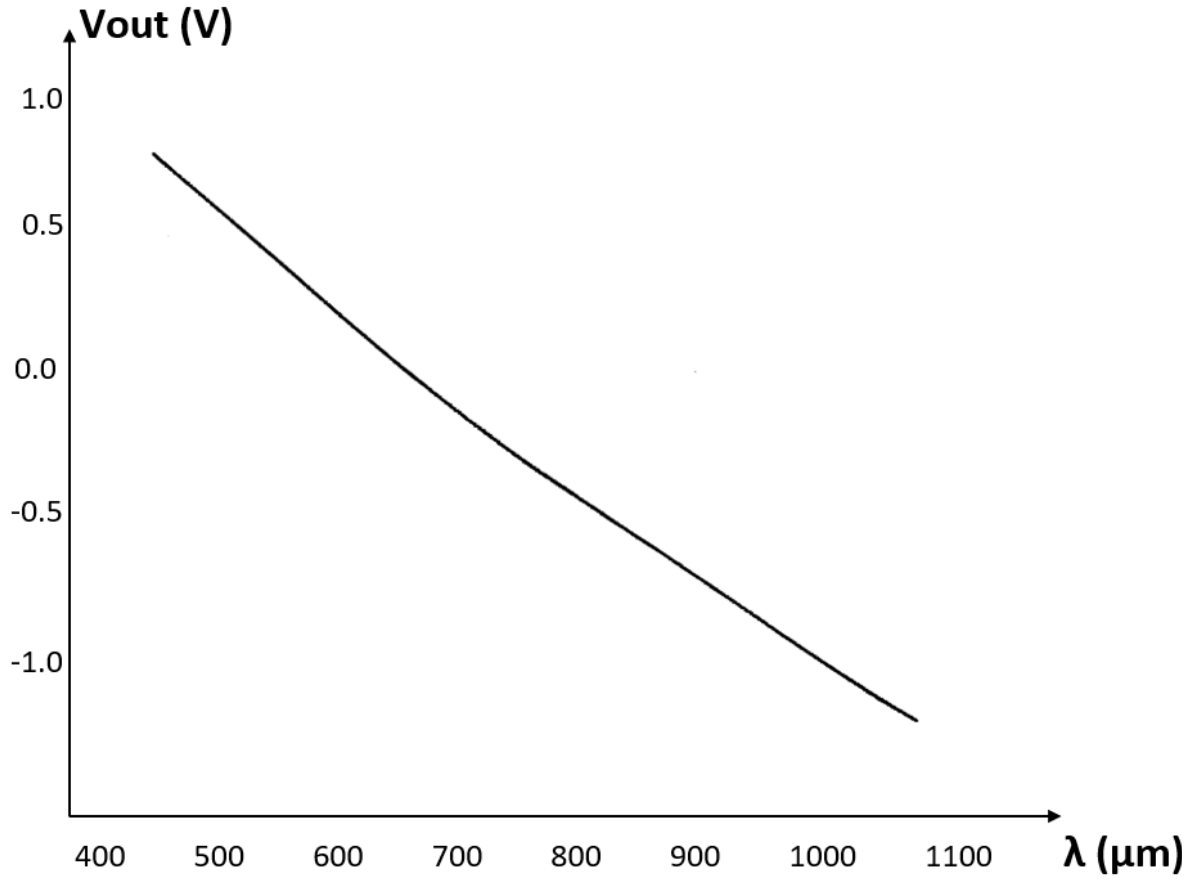
O amplificador logarítmico é usado para relacionar a tensão de saída do amplificador pela corrente de entrada captada pelo fotodetector, respeitando a equação (15) (ASO; KAWANABE, 1982) que nada mais é, do que o comportamento do diodo de realimentação polarizado diretamente. Após as correntes serem amplificadas e correlacionadas pelas tensões de saída dos amplificadores, o circuito subtrator irá efetuar a subtração entre o sinal de saída do amplificador (52) pelo sinal de saída do amplificador (53), que em outras palavras, o resultado matemático será a razão entre a corrente coletada da junção PD_2 pela corrente coletada da junção PD_1, conforme equação (16).

$$I_F \approx \exp\left(\frac{V_F}{KT}\right) \tag{15}$$

$$V_{out} = \mathrm{Log}\left(\frac{I_{PD2}}{I_{PD1}}\right) \tag{16}$$

O resultado da equação (16) é apresentado na Figura 26, onde é possível notar claramente a relação entre a tensão de saída Vout pelo comprimento de onda (λ), ou seja, no circuito subtrator, para uma dada tensão de saída o sistema consegue identificar qual é o comprimento de onda que está sendo detectado. De acordo com (ASO; KAWANABE, 1982) o sinal Vout do circuito subtrator não é influenciado pela intensidade luminosa, devido a detecção do circuito responder somente a variação logarítmica das correntes IPD_2/IPD_1. Já para o circuito somador, o sinal de saída representa a intensidade luminosa captada pelo fotodetector, respeitando a equação (17).

Figura 26 – Vout vs λ do circuito subtrator.



Fonte: ASO; KAWANABE, 1982

$$\text{Log}\,(I_{PD1}.I_{PD2}) \tag{17}$$

4 APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA) dedicado ao desenvolvimento de algoritmos computacionais, permitindo que sistemas adquiriram inteligência para tomada de decisão (BRINK, H.; RICHARDS, 2014). Esta capacidade de aprender é baseada em experiências passadas, que são um conjunto de dados ou instâncias imputados em algoritmos estatísticos que irão tentar encontrar padrões com objetivo de relacionar os dados de entrada com os dados de saída, criando assim, uma inteligência computacional (ROSS et al., 2013; VONRUEDEN et al., 2021).

O AM ainda está longe de ter a mesma inteligência que uma pessoa, entretanto, o mesmo está dando passos largos e significativos rumo a este objetivo. Sua pesquisa concentra-se em processamento de linguagem natural, mecanismos de busca, diagnósticos médicos, bioinformática, reconhecimento de fala, reconhecimento de padrões faciais, reconhecimento de manuscrito, visão computacional, locomoção de robôs e sistemas de previsão (T. G. DIETTRICH, 1997; Y. QIAN, W. ZHOU, J. YAN, W. LI, 2015).

Os algoritmos do AM são divididos em quatro grandes subgrupos, denominados de “Aprendizado Supervisionado”, “Aprendizado Semi-Supervisionado”, “Aprendizado Não-Supervisionado” e “Aprendizado Por Reforço”, conforme Figura 27. Dentro de cada subgrupo estão diversos algoritmos oferecidos pelo AM para que o usuário ou desenvolvedor possa escolher qual o melhor algoritmo a ser utilizado em função do tipo e característica do dado.

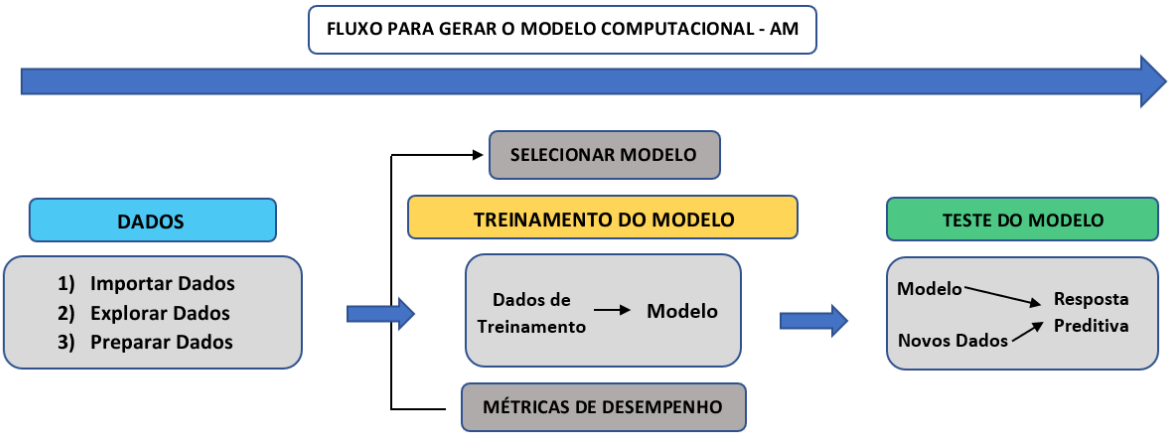
Imagine um sistema capaz de reconhecer as letras do alfabeto escritas a mão por qualquer pessoa. O “grupo de tarefas” tem que possuir a capacidade de reconhecimento e classificação das letras do alfabeto escrita à mão. O “índice de performance do modelo” irá mensurar o percentual de acertos do sistema e a “fonte de dados para o treinamento do modelo”, representa uma base de dados separados contendo uma quantidade de diversas imagens de letras escritas a mão para ensinar o modelo. Sendo assim, a Figura 28 mostra de forma clara e objetiva o processo para gerar um modelo computacional baseado no AM. Inicialmente separa-se uma base significativa de dados para treinar o modelo. O segundo passo é definir qual algoritmo será utilizado em virtude da característica dos dados de treinamento, uma vez definido, inicia-se o processo de treinamento do modelo e por fim, o terceiro passo será testar o modelo para uma classificação ou predição com novos dados que não foram utilizados durante a fase de treinamento. Segundo (A MOSAVI P OZTURK AND K CHAU, [s.d.]), um erro que acontece com frequência nos iniciantes da área do AM, é achar que, pelo fato do modelo ter apresentado ótima eficiência na etapa do treinamento, não significa que essa eficiência será a mesma quando

o modelo for testado para novos dados, ou seja, em hipótese alguma deve-se utilizar os mesmos dados de treinamento para testar o modelo, pois se isso for feito, o modelo não estará aprendendo. Esse tipo de problema é conhecido com *overfitting*, onde o modelo se ajusta perfeitamente na fase de treinamento e torna-se ineficiente para prever novos dados. Para evitar que isso ocorra, segundo (A MOSAVI P OZTURK AND K CHAU, [s.d.]; I O FREEMAN A J HAIGLER S E SCHMEELK L R ELLRODT AND T L, [s.d.]) recomenda-se que dos 100% dos dados, 75% (3/4) são destinados para o treinamento e os 25% (1/4) restantes são para testar o modelo (A MOSAVI P OZTURK AND K CHAU, [s.d.]). Também é utilizado a proporção 2/3 (66,67%) para treinamento e 1/3 (33,33%) para teste (VONRUEDEN et al., 2021; YU et al., 2015).

Figura 27 – Estrutura básica do aprendizado de máquina.

Fonte: Autor

Figura 28 – Fluxo para gerar um modelo computacional baseado no AM.



Fonte: Autor

4.1 APRENDIZADO SUPERVISIONADO

O "Aprendizado Supervisionado" consiste em utilizar uma base da dados, conhecidos como classes ou instâncias, que são previamente classificados, para induzir o modelo a capacidade de prever ou classificar novos dados (classes ou instâncias) de forma precisa, baseado no aprendizado obtido durante a fase de treinamento. Segundo (BRINK, H.; RICHARDS, 2014) a maioria dos problemas para ser solucionados são supervisionados por natureza, logo, é de suma importância que exista um conjunto de dados de treinamento de qualidade, para que o modelo criado possa predizer ou classificar novos dados de forma eficiente (EVERSON, R. M. AND FIELDSEND, 2006; VILARINO, F., KUNCHEVA, L. I., AND RADEVA, 2006).

O primeiro passo para a classificação é mapear e modelar um conjunto de dados composto por uma coleção de exemplos, sendo que cada exemplo é constituído por uma lista de atributos independentes (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). A Figura 29, representa a coleção de exemplos, sendo X_1, X_2, ... X_n os dados de entrada, Y os dados de saída e E_1, E_2, ... E_N as classes ou instâncias. As variáveis x_{11}, x_{12}, y_1, y_2, x_{NM} e y_N são os atributos (características) dos dados de entrada (x_{NM}) e saída (y_N). Como interpretação, a primeira linha possui a classe E_1 que tem os atributos de entrada x_{11}, x_{12} ... x_{1M}, para a saída y_1 e assim por diante.

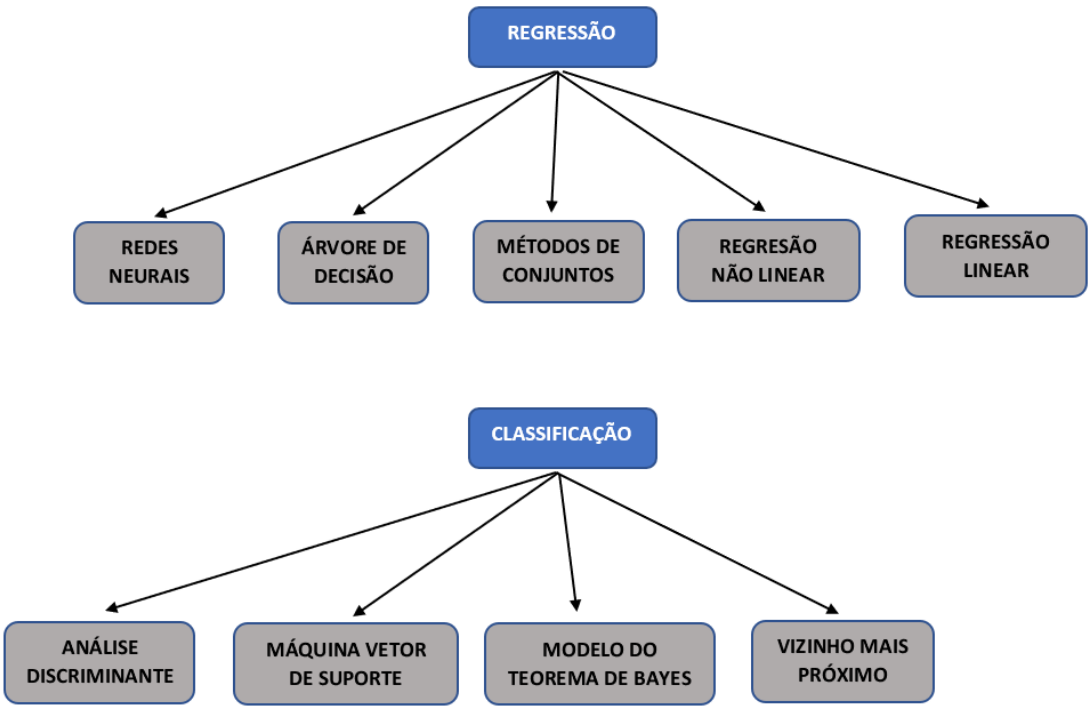
Figura 29 – Conjunto de dados para treinar um modelo no formato classe, atributo e valor.

	X_1	X_2	$\cdots$	X_M	Y
E_1	x_{11}	x_{12}	$\cdots$	x_{1M}	y_1
E_2	x_{21}	x_{22}	$\cdots$	x_{2M}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_N	x_{N1}	x_{N2}	$\cdots$	x_{NM}	y_N

Fonte: HASTIE; TIBSHIRANI; FRIEDMAN, 2009

No aprendizado supervisionado, existem diversos algoritmos que podem ser divididos em dois grandes grupos: REGRESSÃO e CLASSIFICAÇÃO, conforme Figura 30. Para cada grupo existe uma série de algoritmos computacionais usados pelo AM para treinar modelos a partir de uma base de dados (CHAUHAN et al., 2021; LU; LI, 2020). Na Figura 30 é mencionando alguns tipos de algoritmos.

Figura 30 – Algoritmos usados no Aprendizado Supervisionado.



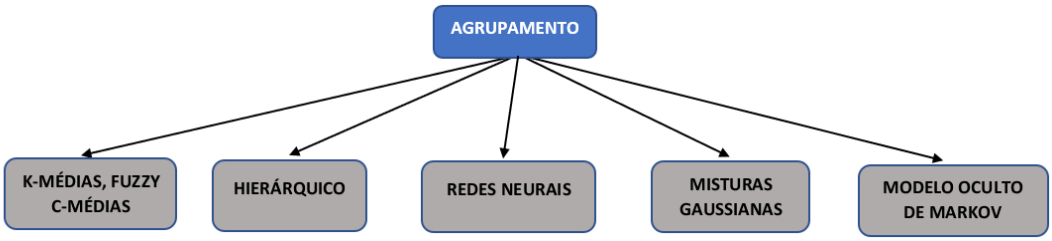
Fonte: Autor

4.2 APRENDIZADO NÃO-SUPERVISIONADO

A proposta do "Aprendizado Não-Supervisionado" a priori não é classificar os dados com modelos preditivos, mas sim, tentar encontrar padrões e regularidades entre os dados de entrada com os dados de saída. Com exemplo, imagine fazer um estudo dos perfis de clientes que compram determinados produtos. Para encontrar esse perfil, inicialmente não se tem muitos dados que possam relacionar o perfil do cliente com determinado produto, logo, os algoritmos de Aprendizado-Não Supervisionado serão responsáveis de criar um modelo que faça essa relação para que o usuário possa entender o comportamento dos dados, que posteriormente será utilizado para classificação (BRINK, H.; RICHARDS, 2014).

A técnica de *Clustering* (Agrupamento) é uma técnica do Aprendizado-Não Supervisionado utilizado para encontrar padrões e classificar uma série de instâncias ou classes de determinado grupo de dados. Na Figura 31 estão dispostos alguns algoritmos de *Cluster* usados no AM. A técnica *Clustering* pode ser entendida como um método de dividir uma certa população de dados em vários grupos, segredando-os e atribuindo-o a cada *cluster*.

Figura 31 – Algoritmos usados no Aprendizado Não-Supervisionado.



Fonte: Autor

4.3 MÉTRICAS DE DESEMPENHO EM APRENDIZADO DE MÁQUINA

Criar um modelo não significa que ele está apto para predizer ou entender o comportamento dos dados. As métricas de desempenho são extremamente importantes para medir a eficácia dos modelos gerados. Nos tópicos abaixo serão tratadas as métricas mais utilizadas.

4.3.1 Matriz de Confusão

A Matriz e Confusão tem o objetivo de entender a relação entre a taxa de acertos e erros que o modelo apresenta em forma de uma matriz, conforme Figura 32. Inicialmente essa matriz é representada por quatro informações, sendo:

- **Verdadeiro Positivo (TP – True Positive):** Representa que no conjunto de dados reais testados, o modelo previu corretamente a classe que está sendo buscada. Como exemplo: Imagina um cientista usando um modelo para classificar pacientes com diabetes, logo, se o paciente está com diabetes o modelo prevê corretamente que este paciente está com diabetes.

- **Falso Positivo (FP – False Positive):** Representa que no conjunto de dados reais testados, o modelo previu erroneamente a classe que está sendo buscada. Como exemplo: Usando a mesma situação no TP, se o paciente não está com diabetes e o modelo prevê erroneamente que o paciente está com diabetes.

- **Falso Verdadeiro (TF – True False):** Representa que no conjunto de dados reais testados, o modelo prevê corretamente a classe que não está sendo buscada. Como exemplo: Usando a mesma situação no TP, se o paciente não está com diabetes e o modelo prevê corretamente que o paciente não está com diabetes.

- **Falso Negativo (NF – Negative False):** Representa que no conjunto de dados reais testados, o modelo prevê erroneamente a classe que não está sendo buscada. Como exemplo: Usando a mesma situação no TP, se o paciente está com diabetes e o modelo prevê erroneamente que o paciente não está com diabetes.

Figura 32 – Matriz de confusão.

		VALORES PREDITOS	
		SIM	NÃO
VALORES REAIS	SIM	VERDADEIRO POSITIVO (TP)	FALSO NEGATIVO (FN)
	NÃO	FALSO POSITIVO (FP)	VERDADEIRO NEGATIVO (TN)

Fonte: Autor

Uma vez entendido a proposta da matriz de confusão, segue abaixo algumas métricas importantes decorrentes da matriz de confusão (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

- ***accuracy* (acurácia):** É uma métrica que diz o quanto o modelo acertou as previsões dentro de um conjunto de dados analisados. Seu cálculo é baseado na razão entre o somatório das previsões corretas (verdadeiros positivos com verdadeiros negativos) pelo somatório de todas as previsões, de acordo com a equação (18).

$$\text{acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

- ***recall* (taxa de positivos verdadeiros):** É uma métrica que indica o quão bom o modelo é em prever os valores positivos dentro de um conjunto de dados analisados. Seu cálculo é baseado na razão entre os verdadeiros positivos pelo somatório dos verdadeiros positivos com os falsos negativos, de acordo com a equação (19).

$$recall = \frac{TP}{TP + FN} \tag{19}$$

- ***precision* (precisão):** É uma métrica que indica o quão eficiente o modelo é, ou seja, qual a proporção de identificações positivas reais que o modelo conseguiu prevê. Seu cálculo é baseado na razão entre os verdadeiros positivos pelo somatório dos verdadeiros positivos com os falsos positivos, de acordo com a equação (20).

$$\text{precisão} = \frac{TP}{TP + FP} \tag{20}$$

- ***f-score:*** É uma métrica que indica o balanço geral entre *recall* e *precision* do modelo gerado. Seu cálculo é baseado na razão do produto pela soma do *recall* com o *precision*, de acordo com a equação (21).

$$\text{f} - \text{score} = 2 * \frac{precision * recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{21}$$

4.3.2 Curva AUC/ROC

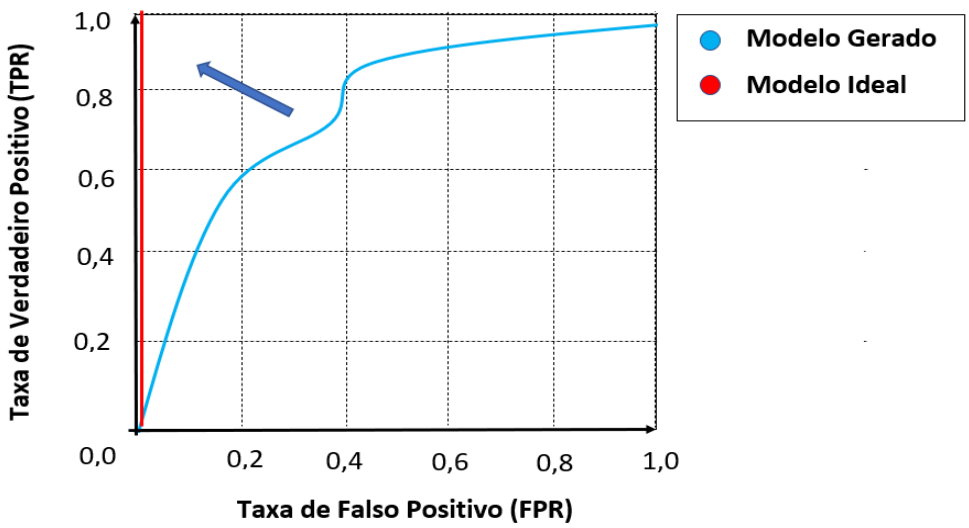
A curva AUC (*Area Under the Curve* – Área Abaixo da Curva) e ROC (*Receiver Operating Characteristics* – Curva de Operação do Receptor) são métricas que indicam o grau de separabilidade entre as os dados classificados, ou seja, permite estudar a variação da "sensibilidade" e "especificidade" dos dados. O cálculo dessa métrica é semelhante a matriz de confusão discutido anteriormente. A curva ROC é um gráfico bidimensional "x" vs "y" num plano designado ROC unitário, onde a taxa de verdadeiro positivo (TPR – *True Positive Rate*) representa a "sensibilidade" e a taxa de falso positivo (FPR – *False Positive Rate*) representa a "especificidade" (Vilarino et al., 2006). Os cálculos de TPR e FPR são mostrados respectivamente nas equações (22) e (23).

$$\text{TPR} = \frac{TP}{TP + FN} \tag{22}$$

$$\text{FPR} = \frac{FP}{TN + FP} \tag{23}$$

A Figura 33 representa a curva ROC para um modelo ideal e um modelo genérico. O modelo ideal está no ponto (0,1) que representa 100% da taxa de verdadeiro positivo e 0% de falso positivo. Já a curva em azul representa um modelo genérico que se aproxima da realidade, possuindo percentuais entre 0% e 100%, indicando que o modelo teve êxito e erros durante a previsão.

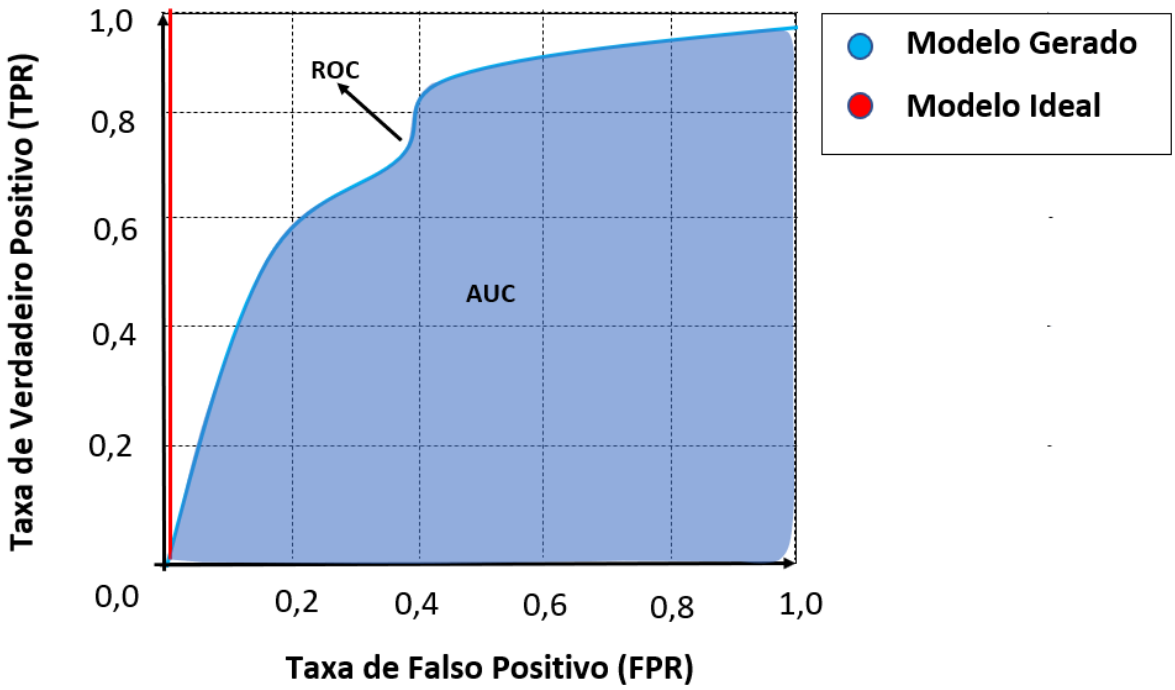
Figura 33 – Curva ROC.



Fonte: Autor

Já a métrica AUC (Figura 34) representa a porção da área de um quadrado unitário abaixo da curva ROC, possuindo valores que vai de 0,0 a 1,0 (HAND, D. J. AND TILL, 2001). Como exemplo, se um classificador tiver AUC = 0,5 significa que o modelo não tem boa capacidade de separabilidade das classes, ou seja, ele só consegue prever 50% dos dados, sendo um modelo ineficiente.

Figura 34 – Curva AUC.



Fonte: Autor

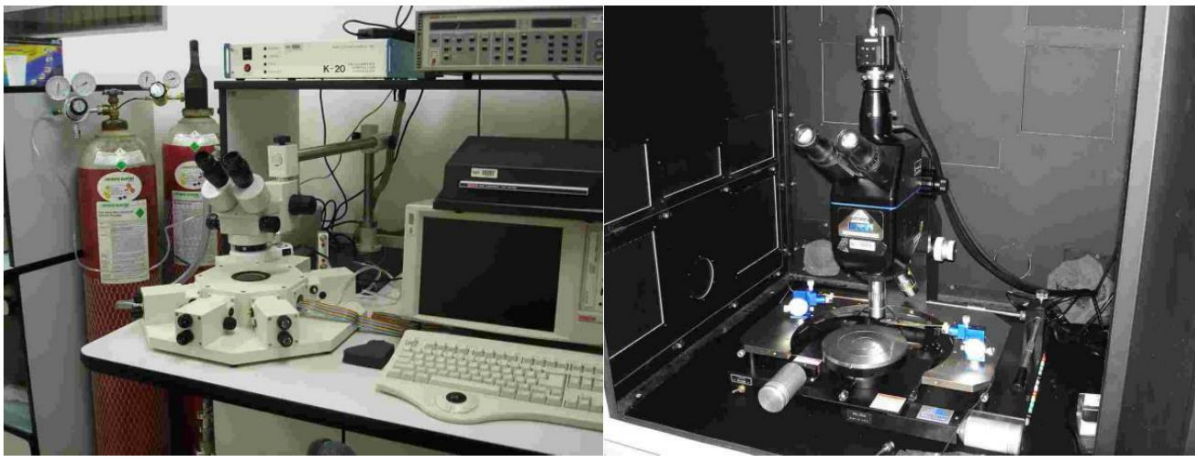
5 MATERIAIS E MÉTODOS

Neste tópico será apresentado o aparato experimental para caracterização do fotodiodo PIN lateral com porta e as ferramentas de simulações Matlab e Atlas utilizadas neste trabalho.

5.1 APARATO EXPERIMENTAL

Para a realização das medidas experimentais foram utilizados os equipamentos Kethley modelo 4200 e Cascade Microtech REL 3600 para caracterização do fotodiodo PIN lateral com porta, conforme Figura 35 – A e 34 – B.

Figura 35 – Fotos dos equipamentos disponíveis na sala de caracterização elétrica do Centro Universitário da FEI: (A) Analisador de parâmetros Keithley 4200, (B) Microprovador Cascade Microtech REL 3600.



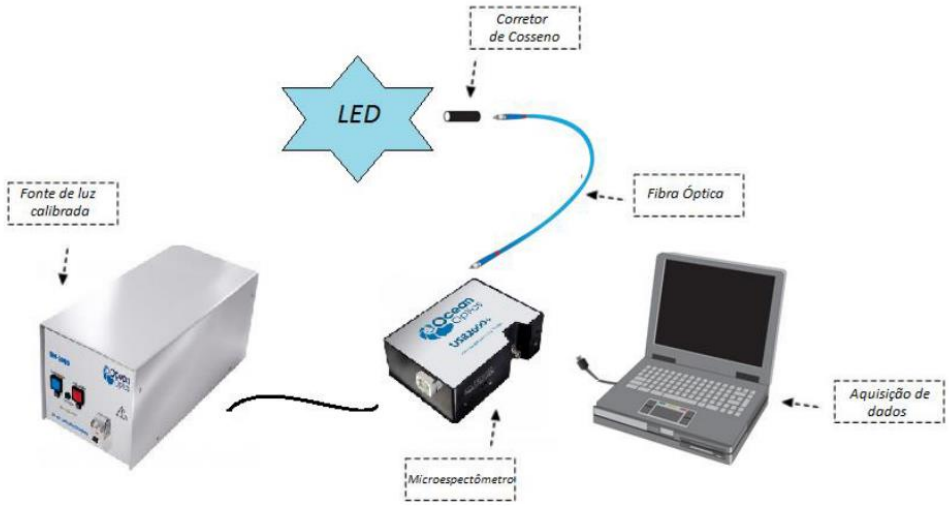
(A) (B)

Fonte: NOVO, 2017

A caracterização dos LEDs é de suma importância para ter a correlação entre a corrente de funcionamento com a potência óptica incidente. Sendo assim, a Figura 36 mostra o esquema de ligação para efetuar a caracterização da potência óptica dos LEDs azul, verde e vermelho através de um microespectômetro que opera na faixa de 350nm a 800nm da empresa *OCEAN OPTICS*, uma fibra óptica e uma fonte de luz de halogênio previamente calibrada do mesmo fabricante (NOVO, 2017). Além disso, foi necessária a utilização de um dispositivo chamado *cosine corrector* (corretor de cosseno), que é um difusor óptico no qual acopla a fibra óptica e

o espectrômetro, com objetivo de captar um sinal com 180 graus de campo de visão (NOVO, 2017).

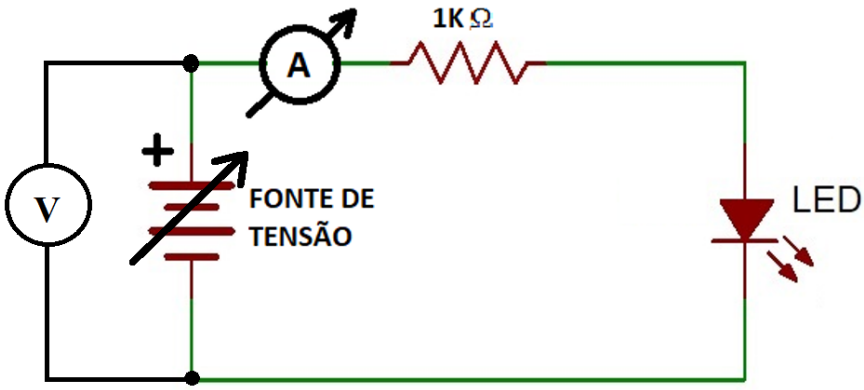
Figura 36 – Esquema de ligação para caracterização da potência óptica incidente dos LEDs.



Fonte: NOVO, 2017

O próximo passo é calibrar a tensão e a corrente que corresponda as potências ópticas caracterizadas. A Figura 37 representa o esquema de ligação, composto de uma fonte de tensão elétrica variável, um voltímetro para medir a tensão elétrica da fonte, um amperímetro para medir a corrente elétrica do Led e um resistor de limitação de corrente de 1KΩ, caso o Led tenha valores elevados de corrente que poderiam causar danos.

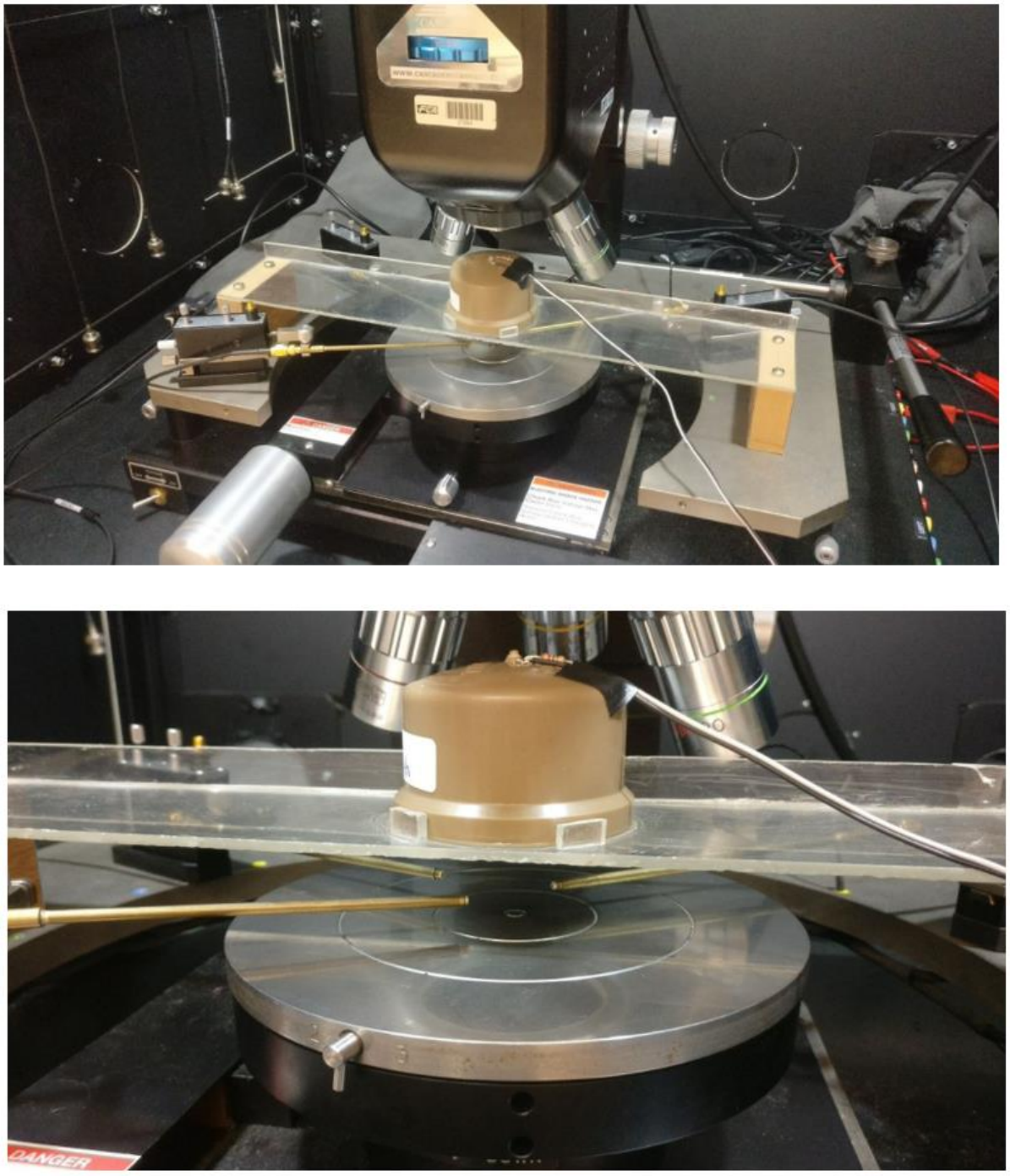
Figura 37 – Esquema de ligação para caracterização da tensão e corrente elétrica dos LEDs.



Fonte: Autor

Um ponto importante a ser mencionado na caracterização do fotodiodo refere-se ao grau de incidência das luzes dos LEDs, que nada mais é do que o espalhamento da luz ao longo do fotodiodo. Para minimizar esta ocorrência, os Leds foram confinados num suporte cilindro (Figura 38) para que toda a potência óptica gerada, seja recebida por toda a área fotossensível do fotodiodo, minimizando o efeito do espalhamento.

Figura 38 – Suporte para minimizar o espalhamento da luz dos Leds.



Fonte: Autor

Por fim, a Tabela 1 mostra o resumo da caracterização dos LEDs, relacionando, comprimento de onda (λ), profundidade de absorção (L_{ABS}), potência óptica incidente, tensão elétrica da fonte e a corrente elétrica do Led.

Tabela 1 – Dados caracterizados dos Leds (experimental).

LED	**λ (µm)**	**L_{ABS} (µm)**	**Potência Óptica (W/m²)**	**Tensão da Fonte (V)**	**Corrente Elétrica (mA)**
Vermelho	0,635	3,06	0,2; 0,4; 0,6	3,3; 4,3; 5,5	8,93; 16,69; 24,69
Verde	0,523	1,14	0,2; 0,4; 0,6	3,7; 4,8; 6,2	5,92; 12,95; 22,25
Azul	0,470	0,581	0,2; 0,4; 0,6	2,7; 2,9; 3,1	1,32; 2,56; 3,88

Fonte: Autor

5.1 O SIMULADOR NUMÉRICO MATLAB

O acrônimo MATLAB (*MAtrix LAboratory*) é um software de computação numérica de alta performance. Seu ambiente de trabalho é muito intuitivo e fácil de se trabalhar através da escrita em linguagem matemática, além da integração da linguagem de programação C, tornando-o extremamente versátil para construção de gráficos, compilação e manipulação de funções, cálculo integro-diferencial, cálculo matricial e operações matemáticas com variáveis simbólicas (MATILAB, [s.d.]). Além disso, o MATLAB, possui uma infinidade de bibliotecas auxiliares, também conhecidas como *Toolbox* (Caixa de Ferramentas) que otimizam o tempo gasto para realizar tarefas, uma vez que, o usuário poderá utilizar funções já definidas, poupando tempo em criá-las. Esse *Toolbox*, possui diversos segmentos, como: *Statistics and Machine Learning* (Estatística e Aprendizado de Máquina), *Curve Fitting* (Ajuste de Curva), *Control System* (Sistema de Controle), *Signal Processing* (Processamento de Sinal), *Mapping* (Mapeamento), *System Identification* (Sistema de Identificação), *Deep Learning* (Aprendizado Profundo), *DSP System* (Sistema Digital de Processamento de Sinal), *Data feed* (*Feed* de Dados), *Financial* (Finanças), *Image Processing* (Processamento de Imagens), *Text Analytics* (Texto Analítico) e *Predictive Maintenance* (Manutenção Preditiva) (MATILAB, [s.d.]). Os modelos computacionais propostos neste trabalho, é desenvolvido no *Classification Learner*, que se encontra na caixa de ferramentas "Estatística e Aprendizado de Máquina" do MATLAB. O *Classification Learner* oferece 22 algoritmos de aprendizado de máquina descritos na Tabela

2. Nessa ferramenta é possível gerar um modelo em linguagem C para efetuar novos treinamentos e prever novos dados.

Tabela 2 – Algoritmos oferecidos pelo *Classification Learner*.

ALGORITMOS
Complex Tree
Medium Tree
Simple Tree
Linear Discriminant
Quadratic Discriminant
Linear SVM
Quadratic SVM
Cubic SVM
Fine Gaussian SVM
Medium Gaussian SVM
Coarse Gaussian SVM
Fine KNN
Medium KNN
Coarse KNN
Cosine KNN
Cubic KNN
Weighted KNN
Ensemble Boosted Trees
Ensemble Bagged Trees
Ensemble Subspace Discriminant
Ensemble Subspace KNN
Ensemble RUSBossted Trees

Fonte: Autor

Uma vez ajustado o modelo que satisfaça as necessidades do usuário, com o código C em mãos, é possível implementá-lo em microprocessadores, microcontroladores e demais softwares, tornando assim as aplicações de predição e classificação extremamente poderosas.

5.2 SIMULADOR DE DISPOSITIVOS ELETRÔNICOS ATLAS

O Atlas é um simulador de dispositivos eletrônicos da renomada empresa Silvaco. O simulador baseia-se num conjunto de equações físicas, permitindo obter uma série de características elétricas associadas às estruturas descritas e as condições de polarizações (ATLAS, [s.d.]). O Atlas fornece simulações bidimensionais e tridimensionais além de simulações de fabricação de processos do dispositivo. Essas opções minimizam grandemente erros de projetos dos dispositivos ou circuitos integrados. Todos os cálculos realizados pelo simulador são obtidos através dos cruzamentos de cada ponto descrito na grade de simulação da estrutura (ATLAS, [s.d.]). Além das curvas características elétricas, é possível determinar variáveis internas como: campo elétrico, densidade de corrente e lacunas, concentração intrínseca de portadores, concentração de elétrons e lacunas, potencial elétrico, taxa de geração e recombinação, entre outros (ATLAS, [s.d.]). Para efetuar uma boa simulação, é de suma importância respeitar três passos importantes no arquivo de simulação, conforme descrito abaixo:

1) Deve-se gerar uma boa grade de simulação, concentrando mais pontos nas regiões onde há mudança de materiais ou concentração de dopantes.
2) Deve-se definir as regiões e os eletrodos do dispositivo.
3) Deve-se definir os modelos físicos pertinentes ao tipo de característica elétrica que se deseja obter.
4) Polarizar o dispositivo e salvar as curvas e estruturas simuladas.

Os modelos utilizados no dispositivo proposto neste trabalho, estão descritos abaixo:

- FERMI: utiliza a característica dos portadores de Fermi-Dirac para solução das equações;

- BGN (*Band Gap Narrowing*): Considera o estreitamento da largura da faixa proibida, sendo importante em regiões com altas concentrações de dopantes;

- KLAASSEN: modelo de mobilidade inicial que considera temperatura, concentração de dopantes e interações entre os portadores;

- FLDMOB (*Parallel Electric Field Dependence*): modelo de degradação de mobilidade que considera o efeito de campo elétrico lateral;

- KLARSH: modelo de geração e recombinação que contabiliza o tempo de vida dos portadores dependentes da concentração de dopantes e da temperatura;

- OPTR: modelo para simulação de dispositivos ópticos, que considera a densidade de estados permitidos para múltiplas transições ópticas;

- KLAAUG (*Klaassen's concentration dependent Auger recombination model*): Modelo e recombinação que considera a transição direta dos portadores;

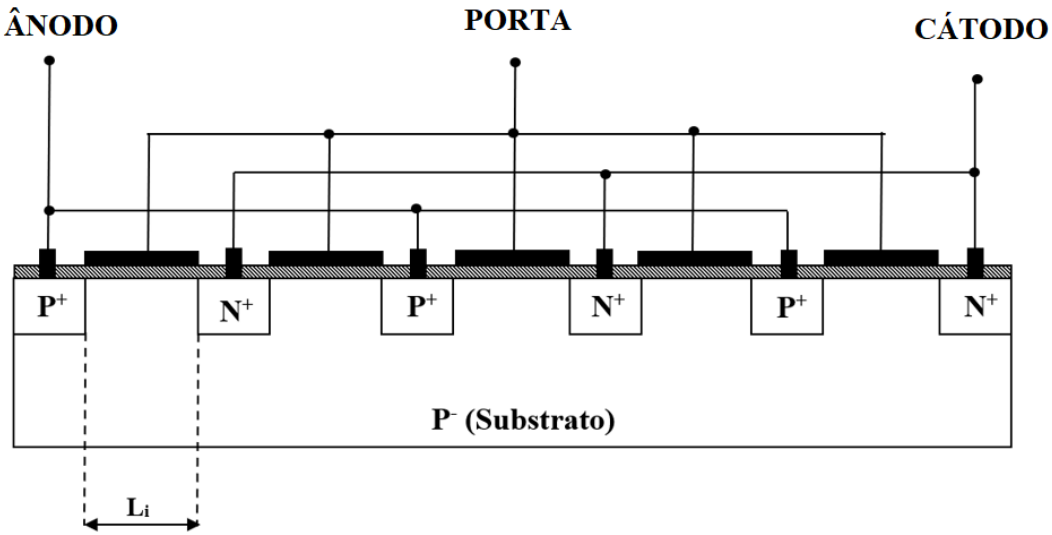
- SHI (*Shirahata*): modelo de degradação da mobilidade pelo campo elétrico transversal e concentração de portadores.

Além dos modelos, foi utilizado nas simulações ópticas o módulo da Atlas conhecido com Luminous, utilizado para calcular o perfil de intensidade óptica absorvida pelo dispositivo e convertê-lo em taxas de fotogeração. Através deste módulo, é possível fazer a discriminação dos comprimentos de ondas incidentes variando 200 a 1000nm. No apêndice A e no apêndice B é descrito o código de simulação utilizado nesta tese.

6 RESULTADOS EXPERIMENTAIS PARA O FOTODIODO PIN LATERAL COM PORTA

Neste tópico serão tratados todos os resultados obtidos pelo fotodiodo PIN lateral com porta fabricado em tecnologia BiCMOS 8HP 0,13µm pelo programa *MOSIS Education Program* (*Mosis Education Program, 2018*). O fotodiodo utilizado é do tipo CMOS, possuindo um perfil de dopagem P+ P- N+, constituído de cinco dedos, sendo que cada dedo é composto das regiões P+ N+ intercaladas pela região P-, o comprimento da região intrínseca L_i é igual a 18µm e toda a região da porta se estende por toda a região intrínseca, conforme Figuras 38.

Figura 39 – Esquema do fotodiodo PIN lateral com porta com 5 dedos.

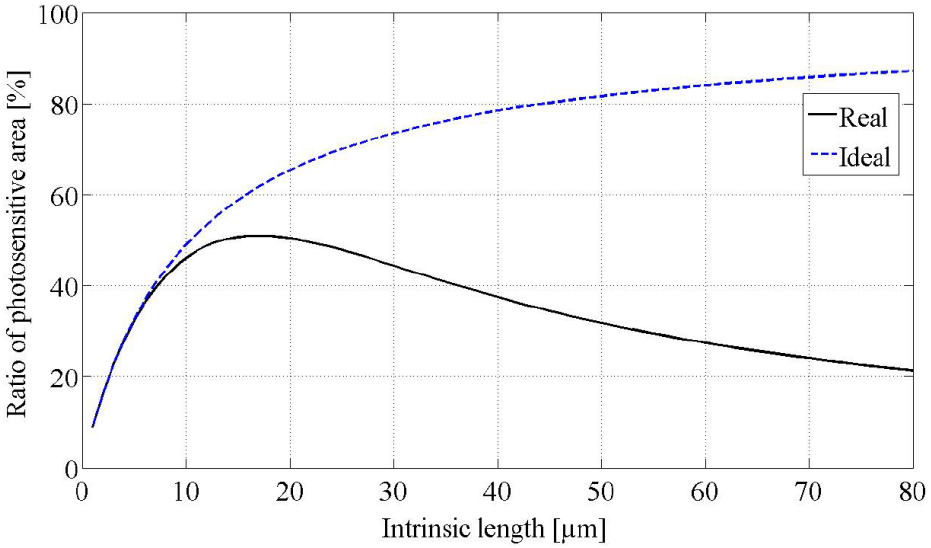


Fonte: Autor

Os dispositivos na configuração multi-dedos, tem por objetivo diminuir os efeitos parasitários e aumentar a sua eficiência quântica, de acordo com (AFZALIAN, 2006; BULTEEL, 2011). Nos projetos de fotodiodos PIN sempre há um grande compromisso em determinar qual o tamanho da região intrínseca, pois ela afeta diretamente a taxa de geração dos pares elétron-lacuna e da corrente de escuro, que por sua vez afetam os índices de mérito discutido em capítulos anteriores. Para extrair a máxima eficiência do fotodiodo, o trabalho proposto por (BULTEEL, 2011) demonstra o equilíbrio (ponto ótimo) entre a corrente de escuro com a corrente fotogerada quando o L_i = 18µm (Figura 40). Por essa razão, a escolha do L_i neste trabalho é baseada no trabalho de (BULTEEL, 2011) para que o fotodiodo PIN lateral com porta possua o máximo de rendimento possível. É importante pontuar que os resultados

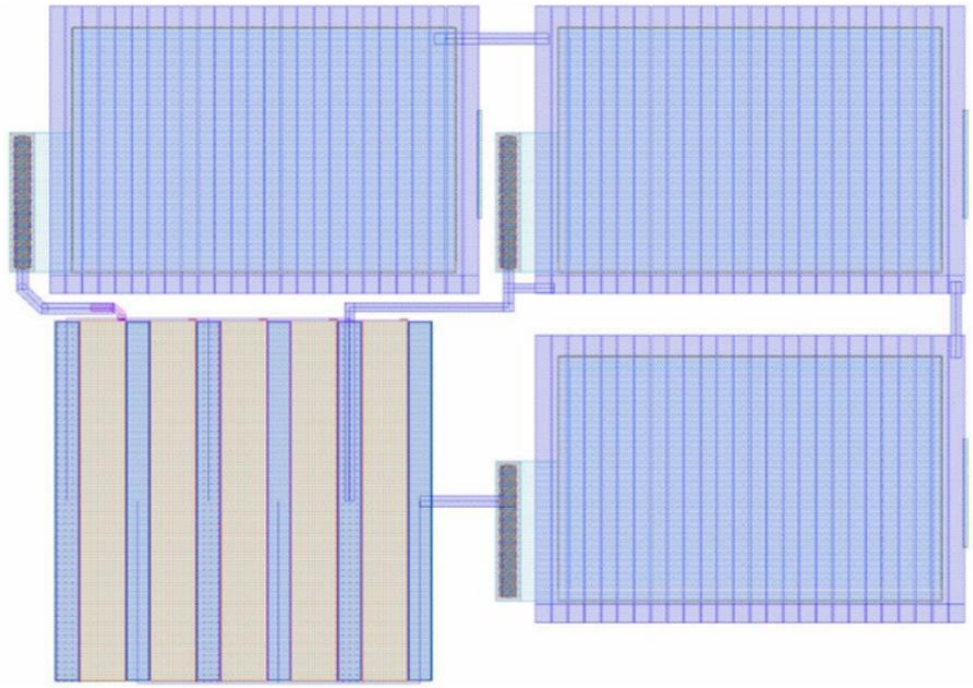
deste trabalho estão focados em discriminar as cores RGB com auxílio do terminal de porta e não no projeto do fotodiodo PIN. A Figura 41 mostra o leiaute real do fotodiodo PIN lateral com porta de cinco dedos utilizado neste trabalho.

Figura 40 – Taxa da área fotossensível vs comprimento intrínseco.



Fonte: BULTEEL, 2011

Figura 41 – Leiaute real do fotodiodo PIN lateral com porta com 5 dedos.

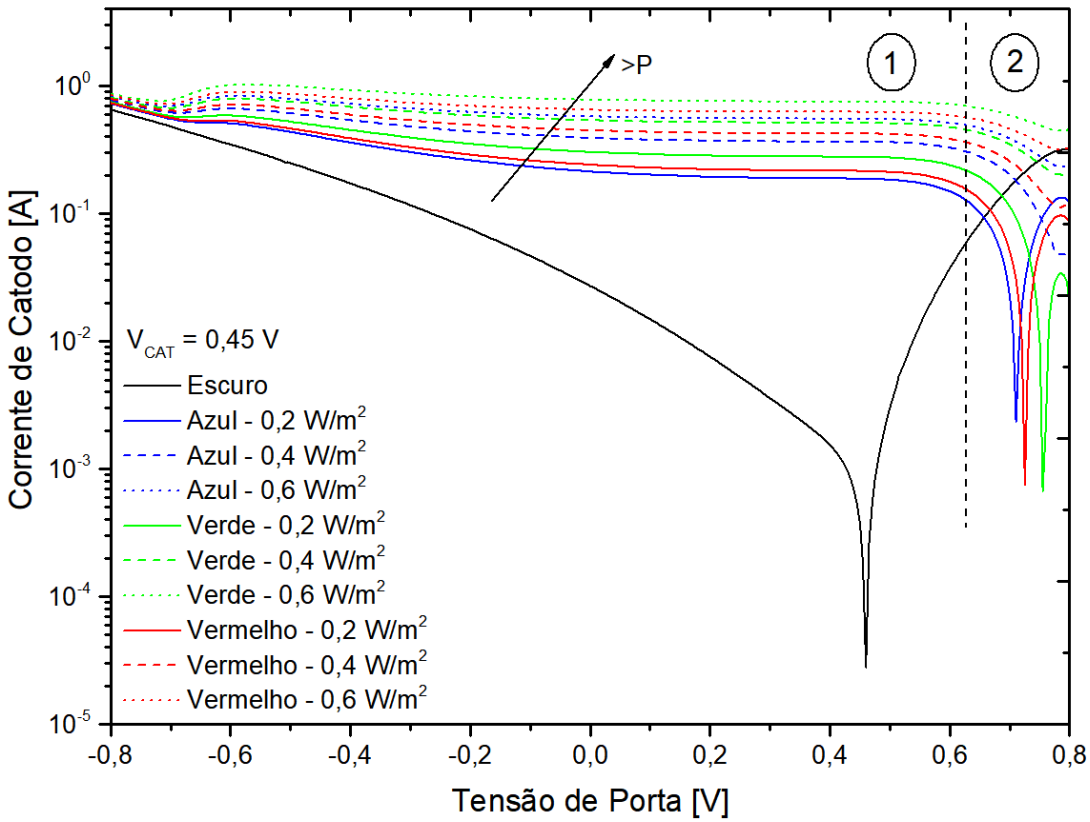


Fonte: Autor

6.1 CARACTERIZAÇÃO DO FOTODIODO PIN LATERAL COM PORTA

Neste tópico será discutido o comportamento da corrente elétrica, tensão elétrica e potência óptica para discriminar as cores RGB com auxílio do terminal de porta. Inicialmente foi feito a caracterização da corrente de catodo pela tensão de porta para as potências de 0,2, 0,4 e 0,6 W/m^2, no intuito de averiguar se o fotodiodo consegue gerar pares elétron-lacuna que irão contribuir para a corrente total no dispositivo. A Figura 42 mostra que o fotodiodo responde bem aos estímulos da incidência óptica para os respectivos comprimentos de ondas, acarretando assim, diferentes níveis de correntes, o que era de se esperar de um fotodetector. A variação da tensão de porta faz com que haja mudança de comportamento do fotodiodo, transitando entre as regiões 1 e 2, que no fundo representa a mudança de estado entre acumulação, depleção e inversão (BAKER, 1997).

Figura 42 – Corrente de catodo (I_{CAT}) pela tensão de porta (V_G) entre -0,8V até 0,8V com polarização reversa de 0,45V (V_{CAT}) (experimental).

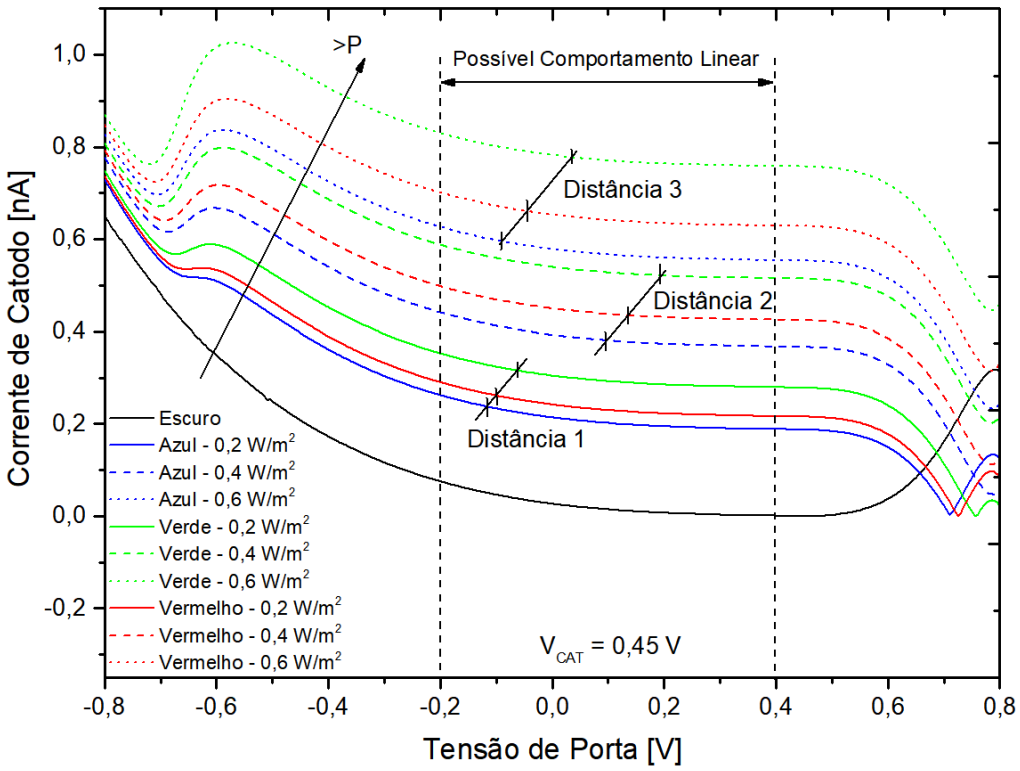


Fonte: Autor

A Figura 43 é a representação linear do gráfico da Figura 42, que mostra alguns comportamentos interessantes. Nota-se que o aumento da potência óptica faz com que a

corrente aumente em virtude da maior geração do par elétron-lacuna, indicada pelas distâncias 1, 2 e 3, o que é bom, dando indícios que os níveis de correntes para os respectivos comprimentos de ondas são bem distintos entre si. Outro ponto importante a ser mencionado é que entre os três comprimentos de onda (vermelho, verde e azul) o verde foi o que apresentou o maior nível de corrente, entrando em contradição, com o discutido em capítulos anteriores, em que, quanto maior for o comprimento de onda, mais baixa é a corrente total devido a menor energia do fóton (SZE, 1981), sendo assim, esperava-se que o comprimento de onda do azul tivesse o maior nível, mas não foi o que ocorreu, pois esse comprimento de onda, é absorvido na superfície do material, fazendo com que haja alta influência da recombinação da superfície, devido as armadilhas de interface e a não periodicidade da rede cristalina (Smart CMOS Image Sensors, [s.d.]). Já o nível de corrente do comprimento de onda do vermelho está dentro do esperado, devido a menor energia do fóton para essa radiação. Além desse ponto, é importante mencionar que na faixa de -0,2V até 0,4V da tensão de porta, as correntes tendem à um possível comportamento linear, o que poderá ser útil para discriminar as cores RGB.

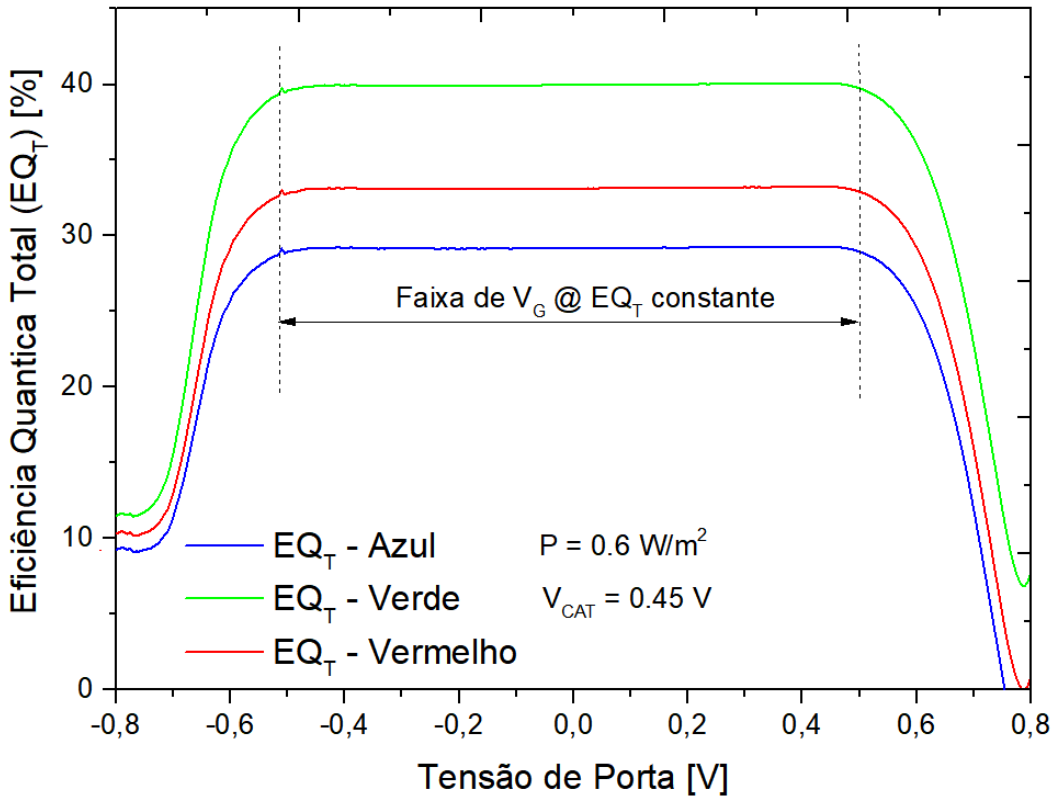
Figura 43 – Corrente de catodo pela tensão de porta entre -0,8V até 0,8V com polarização reversa de 0,45V comportamento linear (experimental).



Fonte: Autor

Outro ponto importante a ser analisado são algumas figuras de mérito para averiguar a eficiência deste dispositivo. A Figura 44 mostra a Eficiência Quântica Total (EQ_T) em função da tensão de porta para potência de 0,6 W/m^2 calculado através da equação (13). A máxima EQ_T para a potência de 0,6 W/m^2 foi alcançada pelo verde com 39,98 %, seguindo do vermelho com 33,23 % e o azul com 29,28 %. Nota-se que para os três comprimentos de onda, a EQ_T se mantém constante para a faixa de tensão de porta $+0{,}5V \leq V_G \leq -0{,}5V$.

Figura 44 – Eficiência Quântica Total (EQ_T) em função da tensão de porta para potência de 0,6 W/m^2 (experimental).

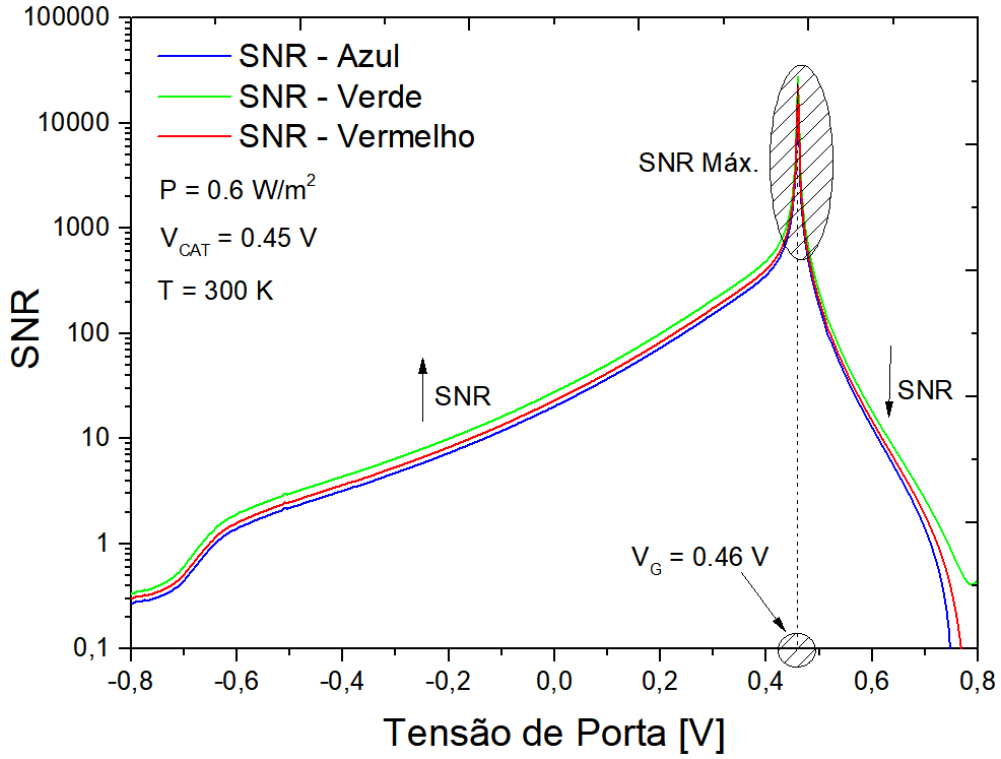


Fonte: Autor

A relação sinal ruído (SNR) em função da tensão de porta deste dispositivo calculado através da equação (11) é apresentado na Figura 45. A máxima SNR alcançada foi para o verde com SNR (dB) = 10.log(27.215,86) = 44,35 dB, seguido do vermelho com SNR (dB) = 10.log(22.586,32) = 43,54 dB e para o azul com SNR (dB) = 10.log(19.890,09) = 42,99. À medida que a tensão de porta aumenta até V_G = 0.46 V a SNR aumenta, pois, a corrente de escuro diminui gradativamente, conforme visto na Figura 43. Quando V_G = 0.46 V, a corrente de escuro é a mínima possível, consequentemente a SNR será a máxima, a partir desse ponto a SNR diminui até V_G = 0.75, pois, no escuro a transição da região de depleção para inversão

ocorre antes quando comparado com as correntes dos respectivos comprimentos de ondas visto na Figura 43.

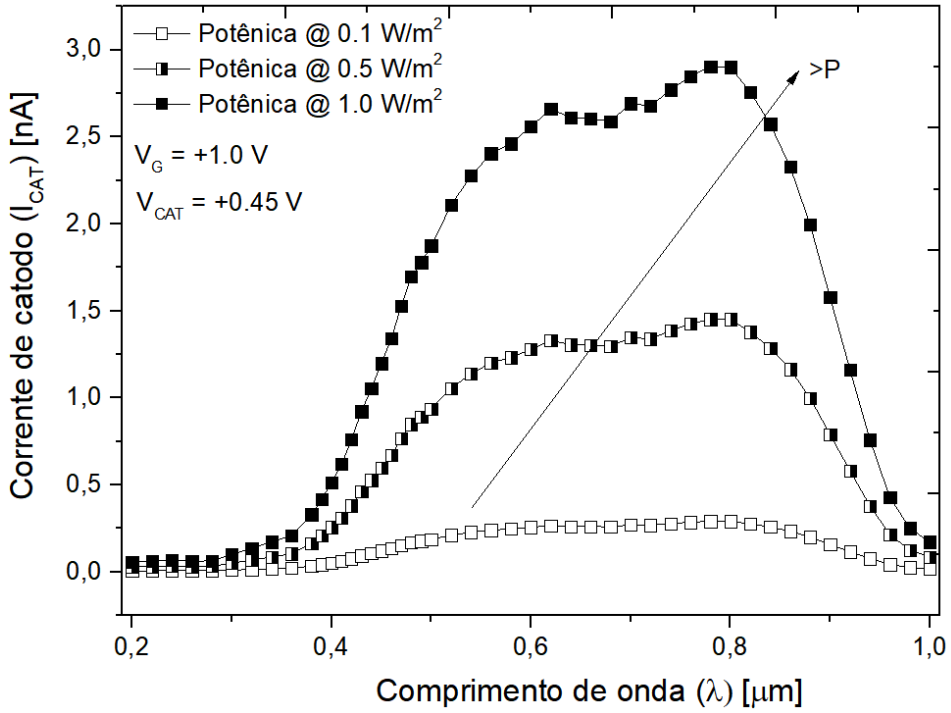
Figura 45 – Relação Sinal Ruído (SNR) em função da tensão de porta para a potência de 0,6 W/m^2 (experimental).



Fonte: Autor

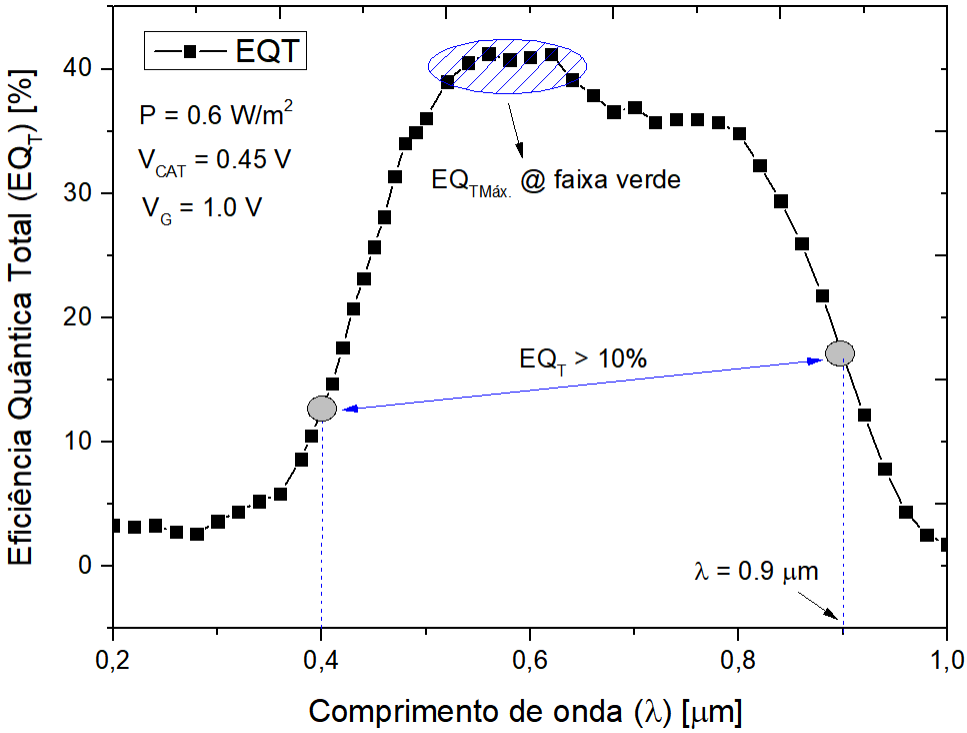
Os LEDs utilizados no experimento possuem limitações de potências, chegando até no máximo 0,6 W/m^2. A fim de explorar ainda mais o comportamento do dispositivo, foi simulado no Atlas a faixa de comprimento de onda 200nm até 1.000nm, para a faixa de potência de 0.1 até 1.2 W/m^2. Na Figura 46 é possível verificar o comportamento da corrente de catodo pelo comprimento de onda com polarização de +1.0 V na tensão de porta para as potências de 0,1; 0,5 e 1 W/m^2. À medida que a potência aumenta, a corrente aumenta para todos os comprimentos de onda em virtude da maior taxa de geração dos pares elétron-lacuna. A faixa de captação dos comprimentos estão entre 340nm à 1.000nm. Outra análise feita, foi verificar a EQ_T dos demais comprimentos de onda, de acordo com a Figura 47. Comprimentos entre 400nm até 900nm possuem EQ_T acima de 10% e a faixa do verde próximo de λ = 500nm possui a máxima EQ_T.

Figura 46 – Corrente de catodo (I_{CAT}) em função do comprimento de onda (λ) (experimental + simulado).



Fonte: Autor

Figura 47 – Eficiência Quântica Total (EQ_T) em função do comprimento de onda (λ) (experimental + simulado).



Fonte: Autor.

6.2 INFLUÊNCIA DA POTÊNCIA ÓPTICA

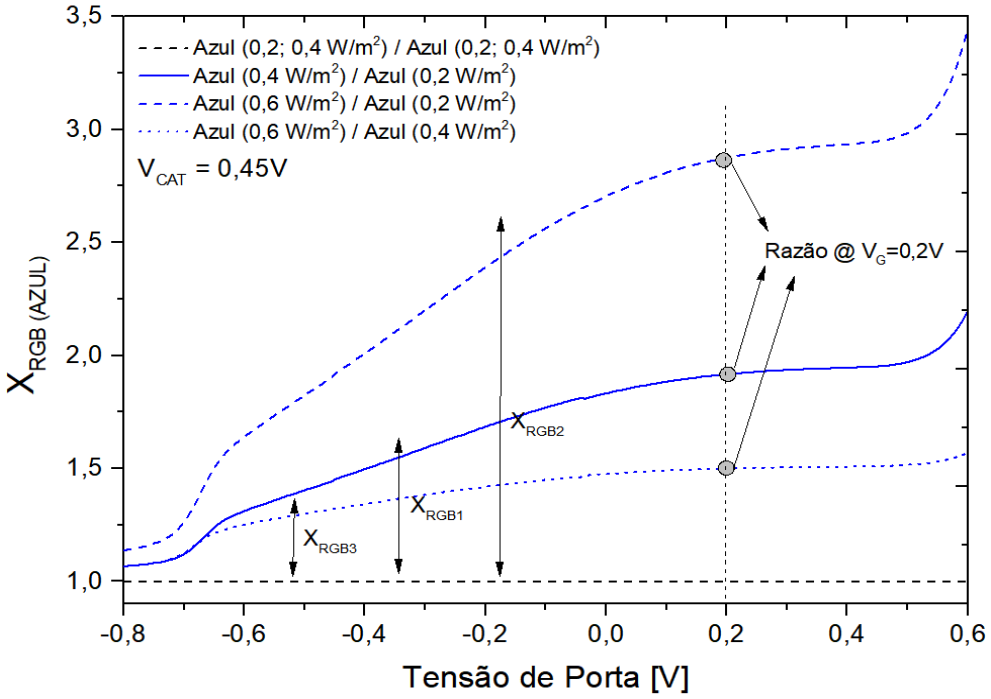
Como a potência óptica influencia bastante no comportamento do fotodiodo, é importante explorar um pouco mais essa interdependência. As Figuras 48, 49 e 50 mostram a razão entre as potências para cada cor. O índice X_{RGB1} representa a razão entre a potência de 0,4 W/m^2 pela potência de 0,2 W/m^2 (equação 24), o índice X_{RGB2} representa a razão entre a potência de 0,6 W/m^2 pela potência de 0,2 W/m^2 (equação 25) e o índice X_{RGB3} representa a razão entre a potência de 0,6 W/m^2 pela potência de 0,4 W/m^2 (equação 26). Essa análise tem como objetivo verificar a sensibilidade do aumento da corrente em função do aumento da potência óptica e da variação da tensão de porta.

$$X_{RGB1} = \frac{I_{CAT(Pot.\ 0{,}4\,W/m^2)}}{I_{CAT(Pot.\ 0{,}2\,W/m^2)}} \tag{24}$$

$$X_{RGB2} = \frac{I_{CAT(Pot.\ 0{,}6\,W/m^2)}}{I_{CAT(Pot.\ 0{,}2\,W/m^2)}} \tag{25}$$

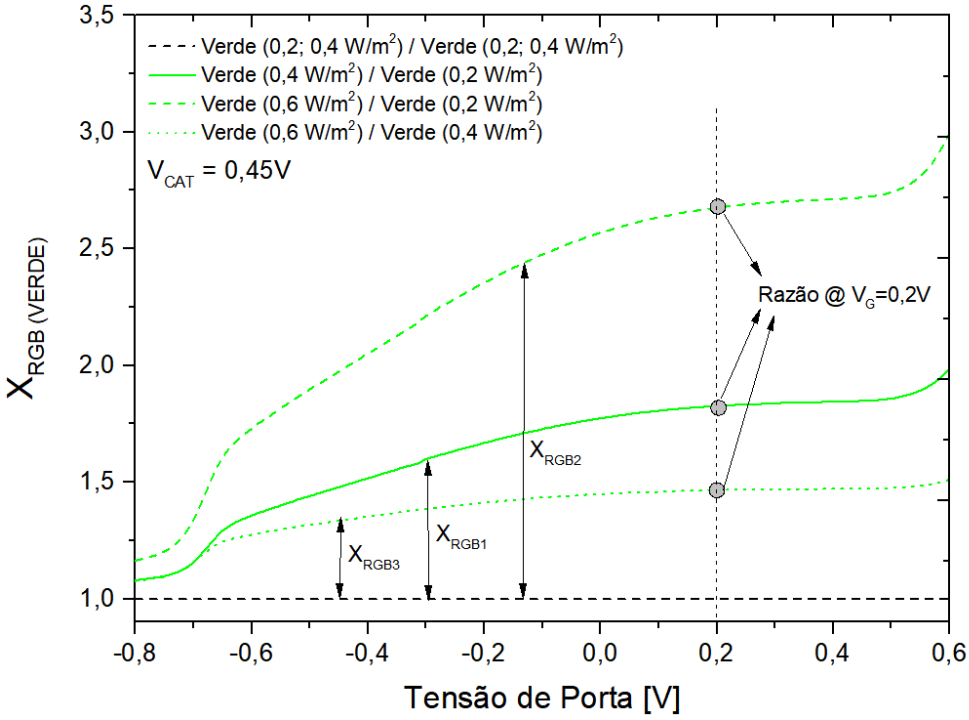
$$X_{RGB3} = \frac{I_{CAT(Pot.\ 0{,}6\,W/m^2)}}{I_{CAT(Pot.\ 0{,}4\,W/m^2)}} \tag{26}$$

Figura 48 – $X_{RGB\ (AZUL)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental).



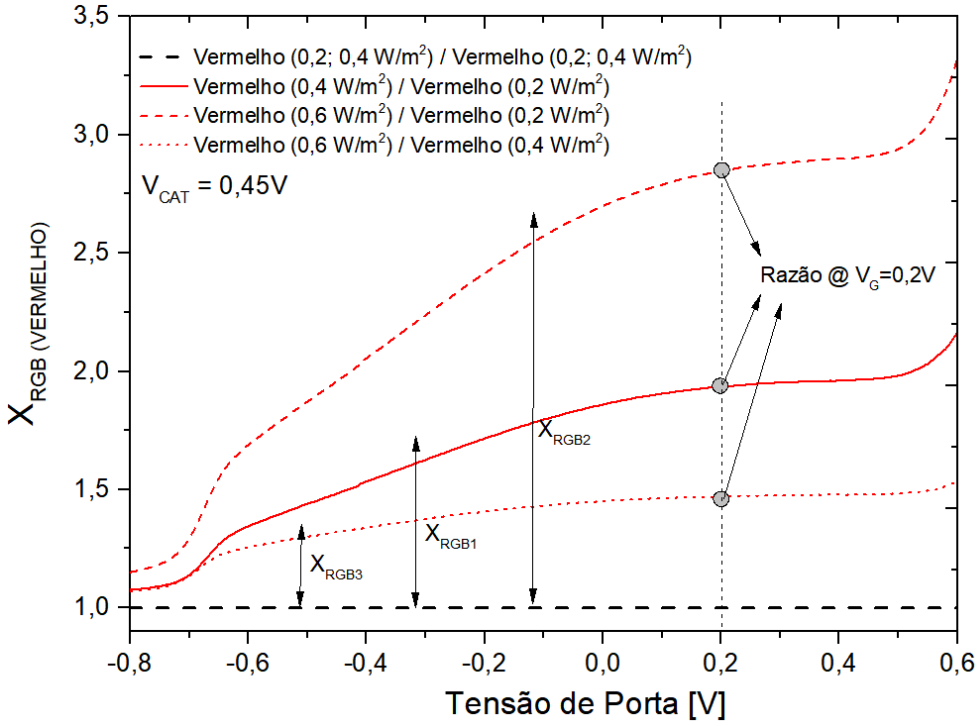
Fonte: Autor

Figura 49 – $X_{RGB\ (VERDE)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental).



Fonte: Autor

Figura 50 – $X_{RGB\ (VERMELHO)}$ pela tensão de porta (V_G) entre -0.8 até 0.6V (experimental).



Fonte: Autor

Nota-se que à medida que a tensão de porta aumenta, a razão $X_{RGB1,2,3}$ também aumenta para todas as cores, evidenciando que o terminal de porta tem um papel fundamental na mudança do comportamento do fotodiodo. Para comparar a sensibilidade das correntes com o aumento da potência óptica de cada cor, foi adotado o ponto de operação de V_G=+0,2V, que possui um dos maiores valores de X_{RGB}. A Tabela 3 mostra os dados extraídos para comparação.

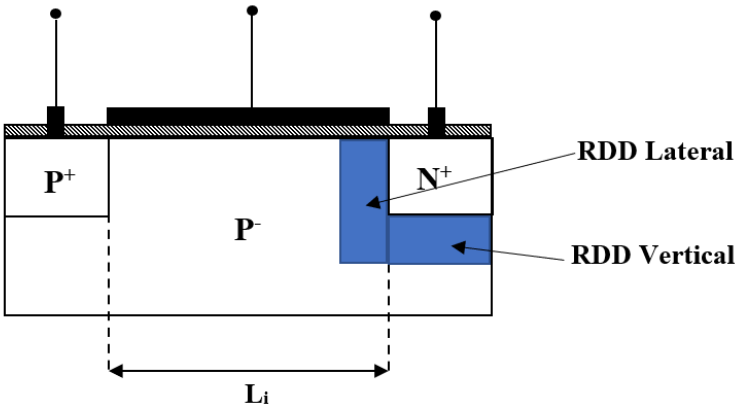
Tabela 3 – Resultados da razão X_{RGB} para V_G=+0,2V (experimental).

LED	Azul	Vermelho	Verde
X_{RGB1}	1,917	1,935	1,826
X_{RGB2}	2,875	2,676	2,676
X_{RGB3}	1,499	1,472	1,466

Fonte: Autor

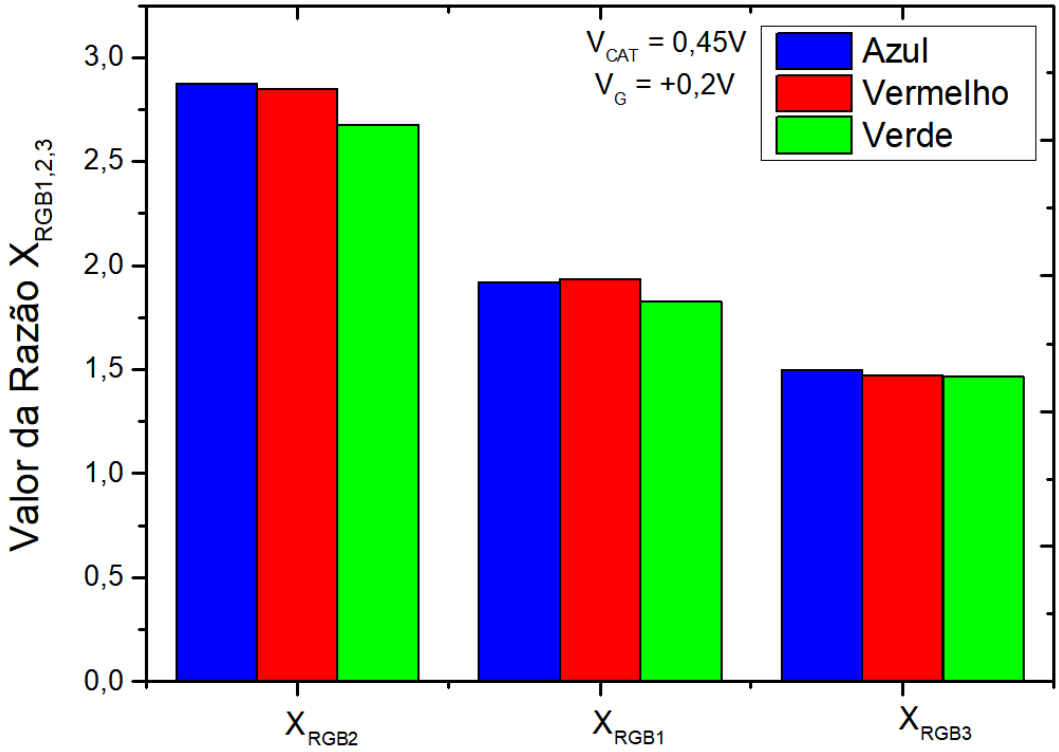
Para interpretação dos dados da Tabela 2, a coluna Azul possui as seguintes informações: A razão X_{RGB1} = 1,917 indica que a corrente da potência de 0,4 W/m^2 é 1,917 vezes maior que a corrente da potência de 0,2 W/m^2. A razão X_{RGB2} = 2,875 indica que a corrente da potência de 0,6 W/m^2 é 2,875 vezes maior que a corrente da potência de 0,2 W/m^2. A razão X_{RGB3} = 1,499 indica que a corrente da potência de 0,6 W/m^2 é 1,499 vezes maior que a corrente da potência de 0,4 W/m^2 e assim por diante para as duas cores restantes. Verifica-se que a cor azul apresentou maior sensibilidade do aumento de potência, seguido do vermelho e por fim para o verde. Para entender esse comportamento deve-se ressaltar que o fotodiodo possui regiões de depleção vertical e lateral, que são parâmetros importantes nas aplicações ópticas, ou seja, a RDD apresenta baixa taxa de recombinação de portadores, conforme esquema da Figura 51. Segundo (SHI, 2012) a profundidade de penetração do silício é pequena para a faixa UV, como o comprimento do azul está próximo dessa faixa, a captação dos portadores gerados é feita próximo da superfície, e é exatamente neste ponto que a tensão de porta irá influenciar de forma positiva na RDD lateral, melhorando a sua captação. O efeito da tensão de porta é similar ao ocorrido nos capacitores CMOS e nos transistores MOS, onde na região de transição entre o óxido e o silício, há uma formação da região de depleção vertical (SZE, 1981). Por isso, o azul teve uma melhor resposta nesta sensibilidade, quando há um aumento da potência. A Figura 52 mostra o resumo comparativo do índice X_{RGB} para todas as cores.

Figura 51 – Região de depleção vertical e lateral do fotodiodo,



Fonte: Autor

Figura 52 – Gráfico de barras X_{RGB} para V_G=+0,2V (experimental).



Fonte: Autor

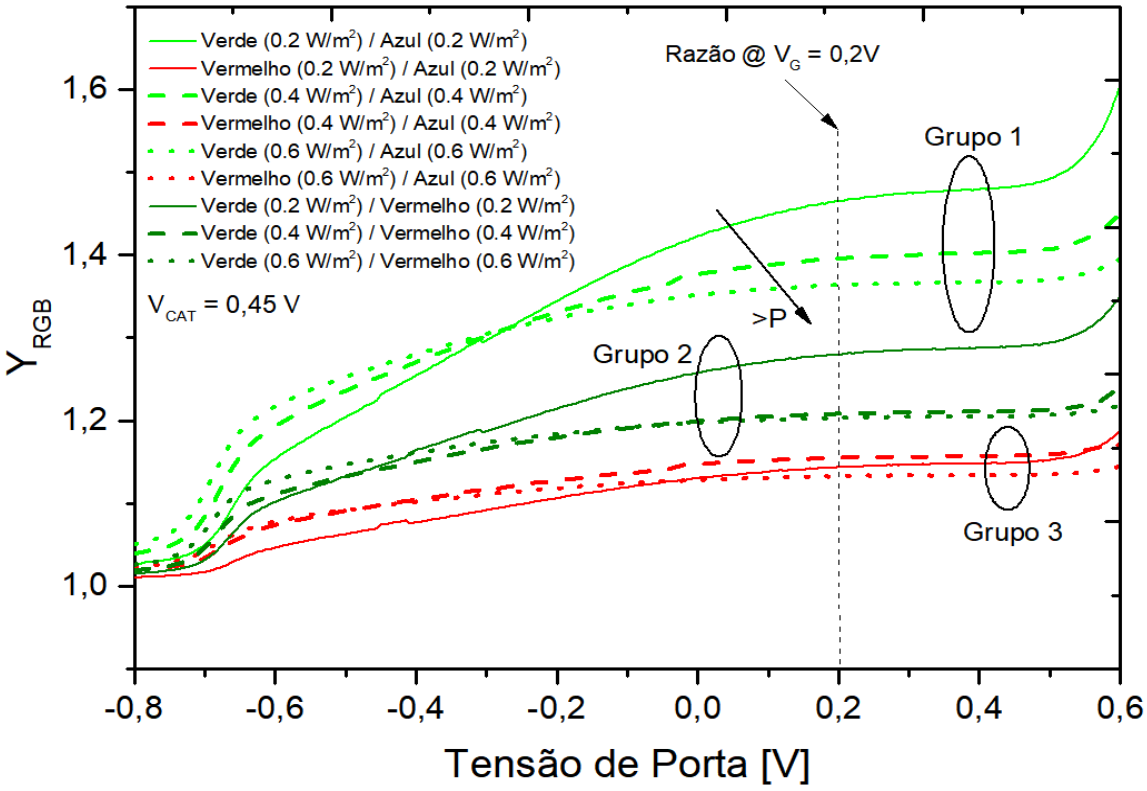
A comparação X_{RGB} de cada cor informa o quanto a corrente é sensível para um aumento de potência. Entretanto a comparação X_{RGB}, não indica o quanto, por exemplo, a corrente da cor azul é maior ou menor em relação a corrente da cor verde, e assim por diante. Portanto, o próximo passo é verificar a razão Y_{RGB} (Figura 53), que representa a comparação dos níveis de

potência entre as cores. Para essa nova análise, é feito as seguintes comparações de acordo com a equação (27):

$$Y_{RGB} = \frac{I_{CAT,Vermelho,Verde(Pot.\ 0,2;\ 0,4;\ 0,6\ W/m^2)}}{I_{CAT,Azul,Vermelho(Pot.\ 0,2;\ 0,4;\ 0,6\ W/m^2)}} \tag{27}$$

- Y_{RGB1} → Verde (0,2 W/m²) pelo Azul (0,2 W/m²);
- Y_{RGB2} → Verde (0,4 W/m²) pelo Azul (0,4 W/m²);
- Y_{RGB3} → Verde (0,6 W/m²) pelo Azul (0,6 W/m²);
- Y_{RGB4} → Vermelho (0,2 W/m²) pelo Azul (0,2 W/m²);
- Y_{RGB5} → Vermelho (0,4 W/m²) pelo Azul (0,4 W/m²);
- Y_{RGB6} → Vermelho (0,6 W/m²) pelo Azul (0,6 W/m²);
- Y_{RGB7} → Verde (0,2 W/m²) pelo Vermelho (0,2 W/m²);
- Y_{RGB8} → Verde (0,4 W/m²) pelo Vermelho (0,4 W/m²);
- Y_{RGB9} → Verde (0,6 W/m²) pelo Vermelho (0,6 W/m²);

Figura 53 – Y_{RGB} vs V_G entre -0.8 até 0.6V (experimental).



Fonte: Autor

Assim como no X_{RGB}, a tensão de porta influencia diretamente no comportamento da corrente do fotodiodo. O Grupo 1 representa a comparação da cor verde pela cor azul, o Grupo 2 representa a comparação da cor verde pela cor vermelho e o Grupo 3 representa a comparação entre a cor vermelha pela cor azul. À medida que a potência aumenta, verifica-se que a razão entre as correntes de cada grupo começa a diminuir, sendo mais evidente no grupo 1. Esse fato ocorre, conforme discutido anteriormente, como a cor azul possui uma maior sensibilidade para a variação da potência, significa que para a mesma variação de potência a cor azul é mais afetada positivamente do que as demais cores, fazendo com que o Y_{RGB} diminua. Para visualizar essa comparação de forma clara e objetiva, a Tabela 4 mostra o resumo dos dados extraídos para a tensão de porta V_G=+0,2V. A Figura 54 mostra o resumo comparativo do índice Y_{RGB} entre as cores para seus respectivos grupos.

Tabela 4 – Resultados da razão Y_{RGB} para V_G=+0,2V (experimental).

LED	Valor	Grupo
Y_{RGB1}	1,46	1
Y_{RGB2}	1,14	3
Y_{RGB3}	1,4	1
Y_{RGB4}	1,15	3
Y_{RGB5}	1,36	1
Y_{RGB6}	1,13	3
Y_{RGB7}	1,28	2
Y_{RGB8}	1,12	2
Y_{RGB9}	1,20	2

Fonte: Autor

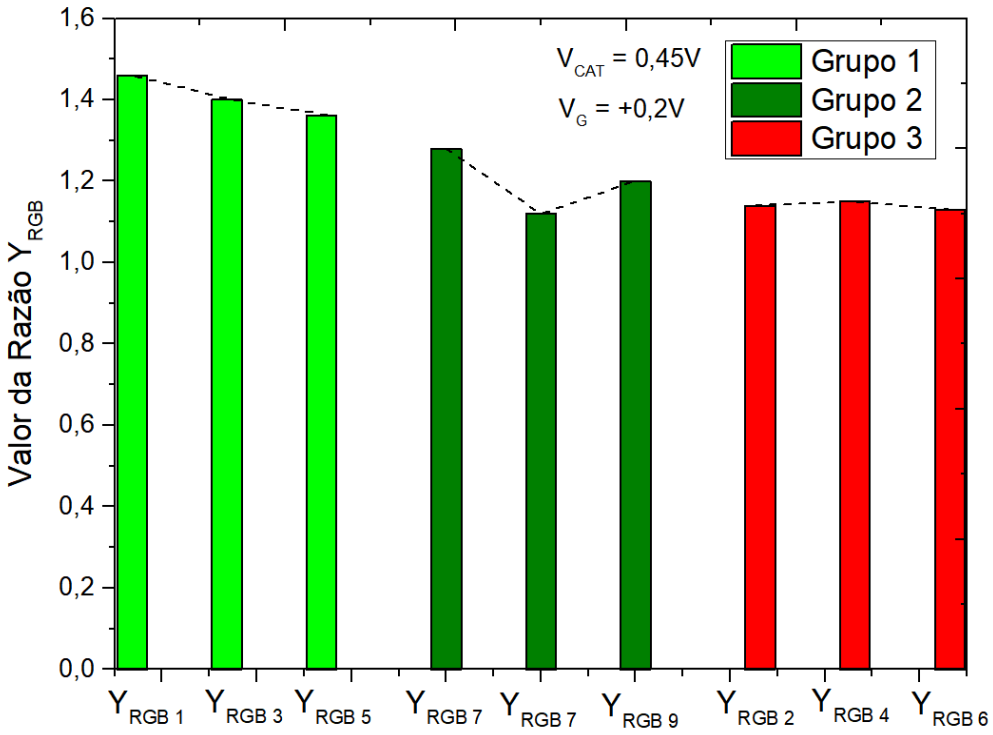
Nota-se na Tabela 3 que os valores entre os índices Y_{RGB} são diferentes entre si, e que a média desse índice está por volta de 1,25, ou seja, significa que em média as cores verde e vermelho são 1,25 vezes maiores que a cor azul (equações 28 e 29) e que a cor verde é 1,25 vezes maior que a cor vermelha (equação 30), tornando-se outro grande potencial para discriminar as cores RGB.

$$I_{VERDE} = 1{,}25.\, I_{AZUL} \tag{28}$$

$$I_{VERMELHO} = 1{,}25.\, I_{AZUL} \tag{29}$$

$$I_{VERDE} = 1{,}25.\, I_{VERMELHO} \tag{30}$$

Figura 54 – Gráfico de barras Y_{RGB} para V_G=+0,2V (experimental).



Fonte: Autor

6.3 USO DA DERIVADA E INTEGRAL PARA DISCRIMINAR AS CORES RGB

Neste tópico serão analisados os métodos derivativos e integrativos sob as curvas de corrente pela tensão de porta para todas as potências ópticas. Para efetuar os cálculos numéricos das derivadas e integrais, este trabalho usa-se do software de análise gráfica Origin Versão 9.1 que em uma das suas caixas de ferramentas disponibiliza algoritmos de aproximação polinomial e cálculos numéricos de equações integro-diferenciais. Inicialmente deseja-se encontra uma relação polinomial entre a corrente de catodo I_{CAT} pela tensão de porta V_G, através da equação (31).

$$I_{CAT}(V_G) = f(V_G) \rightarrow Potêcia(0{,}2; 0{,}4; 0{,}6) \tag{31}$$

Para que os dados possam ser os mais fidedignos com erro residual tendendo a zero, a aproximação polinomial será de nona ordem, conforme equação (32). As Tabelas 5, 6 e 7 mostram os resultados extraídos de cada parâmetro via Origin, sendo:

- bo → Valor que intercepta $I_{CAT}(V_G)$ que representa uma constante;
- B1 à B9 → Parâmetros da equação;
- ERRO → Representa a diferença entre o dado real e o dado aproximado pelo polinômio;

$$I_{CAT}(V_G) = {V_G}^9.B_9 + {V_G}^8.B_8 + {V_G}^7.B_7 + {V_G}^6.B_6 + {V_G}^5.B_5 + {V_G}^4.B_4 + {V_G}^3.B_3 + {V_G}^2.B_2 + {V_G}^1.B_1 + b_0 \quad (32)$$

Nota-se nas tabelas que os parâmetros se diferem entre si, o que é um bom sinal para determinar qualquer ponto de trabalho através da coordenada cartesiana (V_G, I_{CAT}). O ERRO apresenta em média uma diferença entre o dado real e o aproximado na ordem de grandeza de 10^{-11}, indicando uma aproximação de praticamente 100%.

Tabela 5 – Aproximação polinomial para o azul (experimental).

PARÂMETROS	**0,2 W/m²**		**0,4 W/m²**		**0,6 W/m²**	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
bo	2,14E-10	2,70E-14	3,93E-10	3,55E-14	5,79E-10	2,52E-14
B1	-1,53E-10	2,85E-13	-1,58E-10	3,74E-13	-1,50E-10	2,66E-13
B2	3,68E-10	1,46E-12	3,82E-10	1,92E-12	3,72E-10	1,37E-12
B3	-4,06E-10	8,91E-12	-3,06E-10	1,17E-11	-4,10E-10	8,32E-12
B4	-2,20E-10	1,96E-11	-3,74E-10	2,57E-11	-3,20E-10	1,83E-11
B5	3,67E-10	8,81E-11	-4,08E-10	1,16E-10	4,32E-10	8,23E-11
B6	3,21E-09	8,97E-11	4,00E-09	1,18E-10	4,18E-09	8,38E-11
B7	-1,93E-09	3,37E-10	5,08E-10	4,42E-10	-2,81E-09	3,15E-10
B8	-8,24E-09	1,31E-10	-1,01E-08	1,72E-10	-1,14E-08	1,23E-10
B9	2,42E-09	4,35E-10	7,79E-10	5,72E-10	6,28E-09	4,07E-10

Fonte: Autor

Tabela 6 – Aproximação polinomial para o verde (experimental).

PARÂMETROS	0,2 W/m²		0,4 W/m²		0,6 W/m²	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
bo	3,05E-10	4,34E-14	5,41E-10	2,45E-14	6,54E-10	2,55E-14
B1	-1,51E-10	4,57E-13	-1,51E-10	2,58E-13	-1,50E-10	2,69E-13
B2	3,61E-10	2,35E-12	3,72E-10	1,33E-12	3,76E-10	1,38E-12
B3	-4,34E-10	1,43E-11	-4,04E-10	8,07E-12	-4,12E-10	8,42E-12
B4	-2,98E-11	3,14E-11	-3,02E-10	1,77E-11	-4,04E-10	1,85E-11
B5	3,14E-10	1,42E-10	3,72E-10	7,98E-11	5,00E-10	8,33E-11
B6	2,38E-09	1,44E-10	4,06E-09	8,12E-11	4,75E-09	8,48E-11
B7	-1,08E-09	5,41E-10	-2,54E-09	3,05E-10	-3,39E-09	3,18E-10
B8	-7,55E-09	2,11E-10	-1,11E-08	1,19E-10	-1,26E-08	1,24E-10
B9	1,52E-09	6,99E-10	5,83E-09	3,94E-10	7,82E-09	4,11E-10

Fonte: Autor

Tabela 7 – Aproximação polinomial para vermelho (experimental).

PARÂMETROS	0,2 W/m²		0,4 W/m²		0,6 W/m²	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
bo	2,42E-10	3,29E-14	4,51E-10	3,07E-14	6,54E-10	2,55E-14
B1	-1,53E-10	3,47E-13	-1,51E-10	3,24E-13	-1,50E-10	2,69E-13
B2	3,64E-10	1,78E-12	3,73E-10	1,67E-12	3,76E-10	1,38E-12
B3	-3,70E-10	1,08E-11	-3,87E-10	1,01E-11	-4,12E-10	8,42E-12
B4	-1,54E-10	2,38E-11	-3,25E-10	2,22E-11	-4,04E-10	1,85E-11
B5	-9,50E-11	1,07E-10	8,07E-11	1,00E-10	5,00E-10	8,33E-11
B6	2,99E-09	1,09E-10	4,08E-09	1,02E-10	4,75E-09	8,48E-11
B7	-8,24E-11	4,10E-10	-8,42E-10	3,83E-10	-3,39E-09	3,18E-10
B8	-8,17E-09	1,60E-10	-1,08E-08	1,49E-10	-1,26E-08	1,24E-10
B9	3,07E-10	5,30E-10	2,43E-09	4,95E-10	7,82E-09	4,11E-10

Fonte: Autor

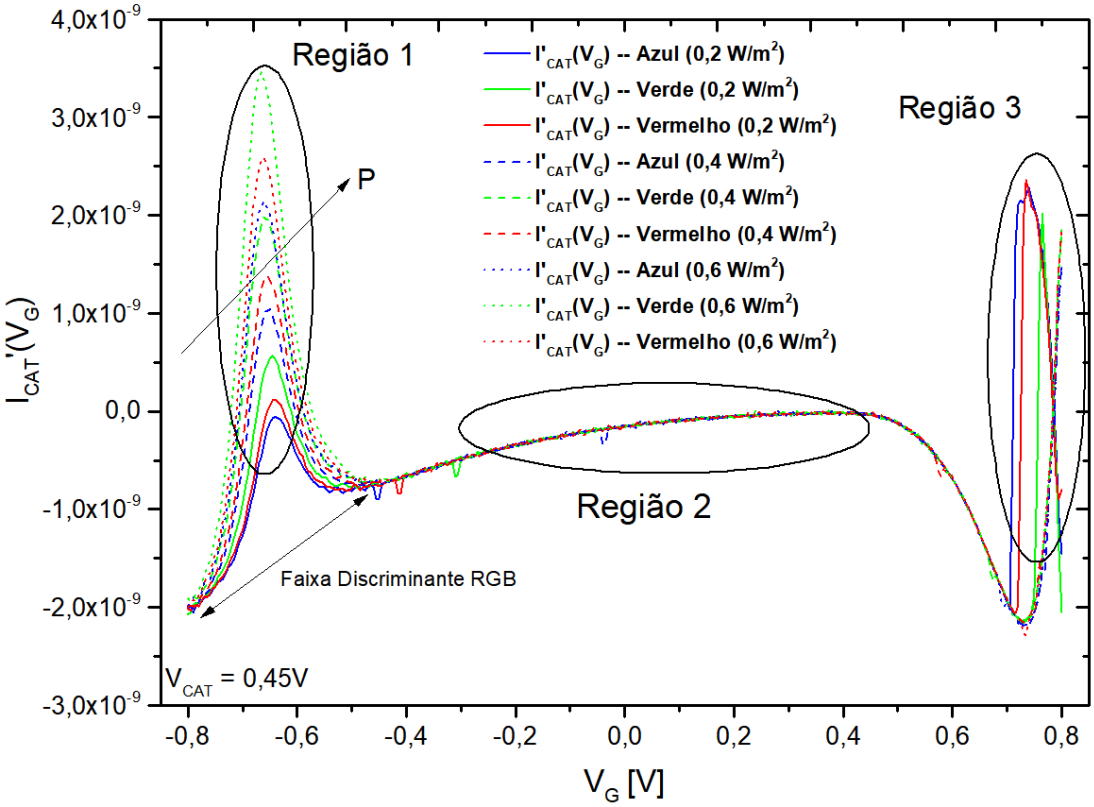
6.3.1 Método Derivacional

Tendo em mãos as equações polinomiais das cores, o próximo passo e efetuar a derivada da corrente pela tensão de porta, conforme equação (33).

$$I'_{CAT}(V_G) = \frac{dI_{CAT}(V_G)}{dV_G} \tag{33}$$

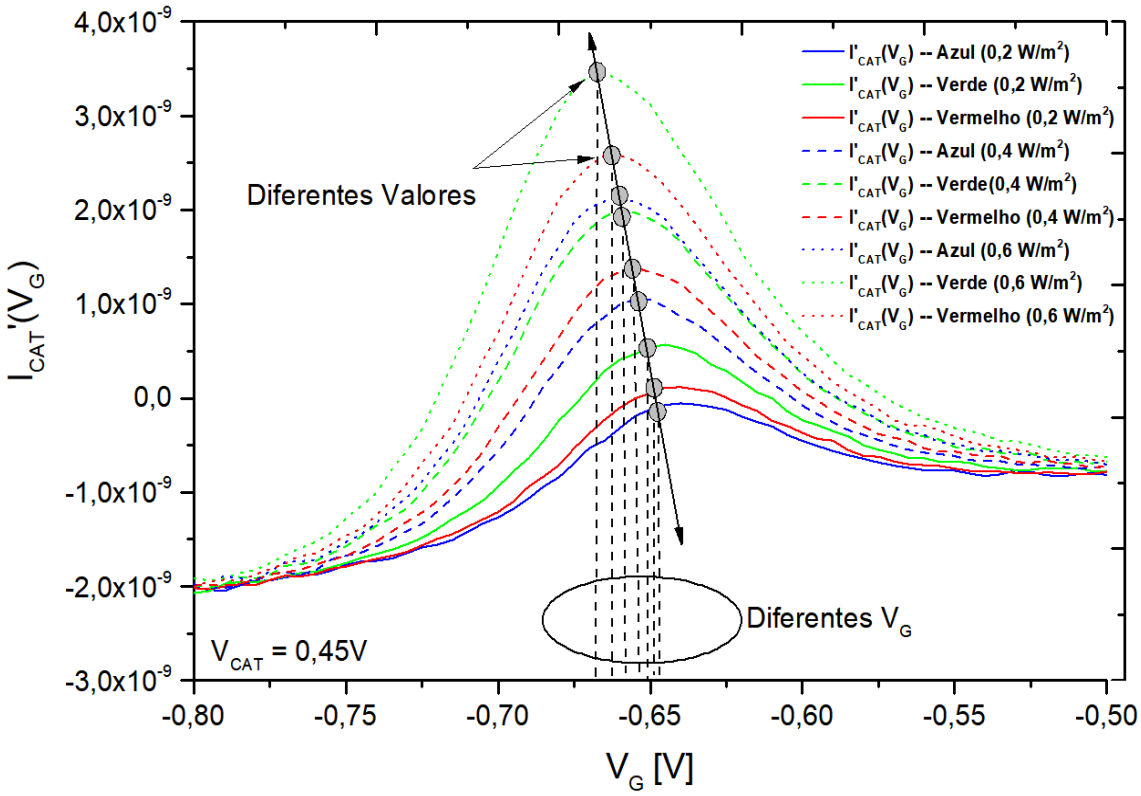
A Figura 55 mostra o cálculo da derivada para as cores RGB. Verifica-se que na região "1" compreendida entre a faixa de -0,8V à 0,5V apresenta uma possível discriminação das cores através dos picos distintos gerados pelo resultado da derivada. As regiões "2" e "3" não apresentam nenhum comportamento relevante, pois os dados são sobrepostos uns aos outros. A fim de investigar ainda mais a região "1", a Figura 56 mostra o resultado expandido da faixa -0,8V à -0,5V a fim de analisar em detalhes o seu comportamento. Os círculos em cinza informam os pontos máximos do resultado da derivada para valores diferentes de V_G, conforme dados extraídos na Tabela 8.

Figura 55 – Gráfico de $I'_{CAT}(V_G)$ x V_G para V_{CAT} = 0,45V (experimental).



Fonte: Autor

Figura 56 – Gráfico expandido da faixa -0,8V à -0,5V de I'$_{CAT}$(V_G) x V_G para V_{CAT} = 0,45V (experimental).



Fonte: Autor

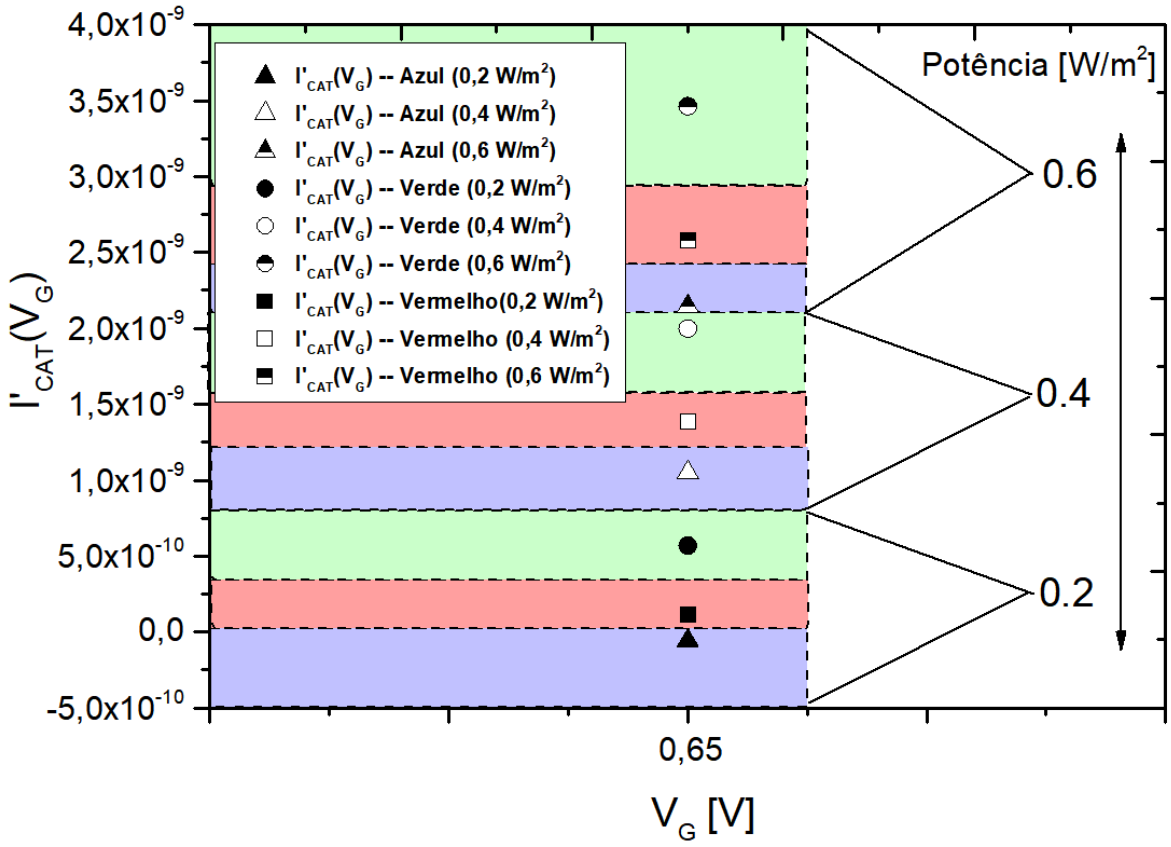
Tabela 8 – Dados extraídos do pico máximo da derivada (experimental).

Cores	Potência [W/m²]	V_G [V]	I'$_{CAT}$(V_G) [A/V]
Azul	0.2	-0,64	-5,64E-11
	0.4	-0,655	1,05E-09
	0.6	-0,66	2,14E-09
Verde	0.2	-0,645	5,69E-10
	0.4	-0,66	2,00E-09
	0.6	-0,665	3,46E-09
Vermelho	0.2	-0,64	1,18E-10
	0.4	-0,655	1,39E-09
	0.6	-0,665	2,58E-09

Fonte: Autor

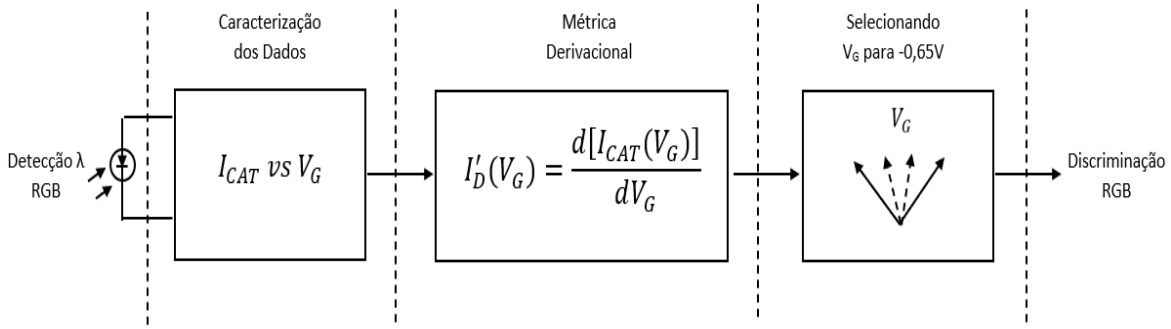
Através da tabela 7 é possível verificar que a faixa de V_G para o ponto máximo da derivada é muito pequena, logo, a média destes pontos estão em torno de -0,65V. Sendo assim, Figura 57 mostra via gráfico os resultados da tabela 7, indicando que para os três níveis de potência óptica, é possível discriminar a radiação RGB para V_G = -0,65V. A Figura 58 mostra o diagrama de blocos passo a passo desse método.

Figura 57 – Gráfico de $I'_{CAT}(V_G)$ x V_G para V_G = 0,65V (experimental).



Fonte: Autor

Figura 58 – Diagrama de blocos do método derivacional para discriminação RGB.



Fonte: Autor

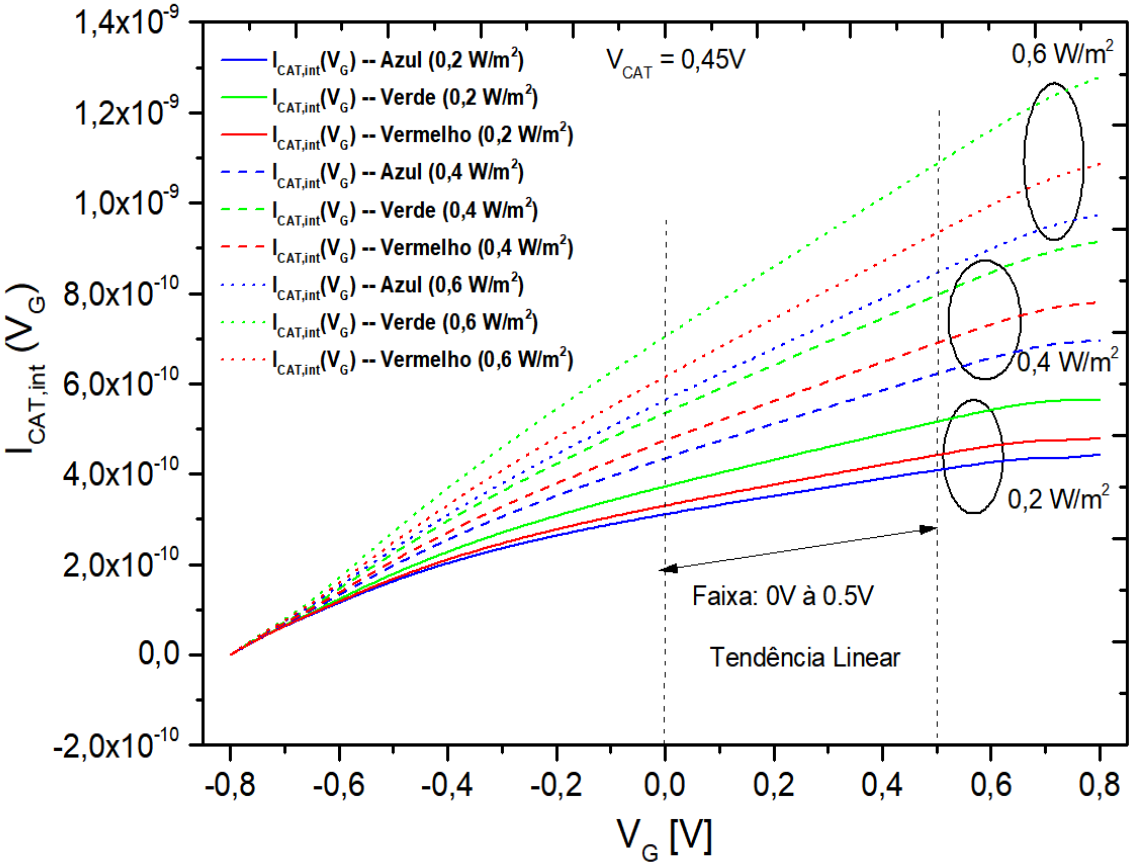
6.3.2 Método Integrativo

Assim como o método derivativo foi eficaz para discriminar as cores RGB, está seção irá tratar as análises do método integrativo nos dados de corrente de catodo do fotodiodo pela tensão de porta. Tendo em mãos a aproximação polinomial das cores, a equação (34) executa o cálculo da integral.

$$I_{CAT,int.}(V_G) = \int_{-V_G}^{+V_G} I_{CAT}(V_G).\,dV_G \tag{34}$$

Sabe-se que o resultado da integral, nada mais é do que o cálculo de toda a área que está abaixo das curvas I_{CAT} vs V_G do fotodiodo. A Figura 59 representa esse cálculo e trazendo consigo comportamentos extremamente promissores para discriminação RGB.

Figura 59 – Gráfico de $I_{CAT,int}(V_G)$ x V_G para V_{CAT} = 0,45V (experimental).



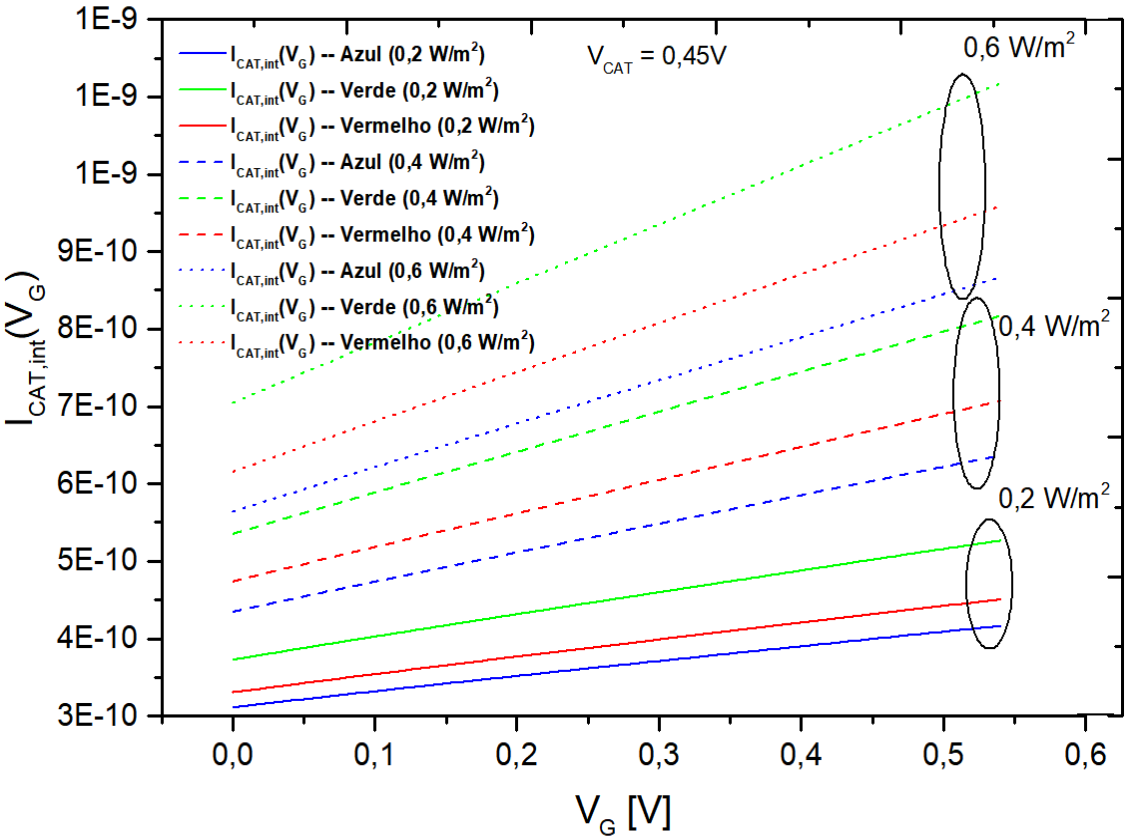
Fonte: Autor

Nota-se que à medida que a tensão de porta vai aumentando o resultado da integral de cada curva vai se distanciando uma das outras com uma tendência linear. Esse comportamento é extremamente promissor, pois fazendo uma aproximação linear de uma dada região de V_G, o coeficiente angular de cada cor para a sua respectiva potência será o índice discriminante RGB. A Figura 60 mostra os dados expandidos da faixa de 0V à 0,5V da tensão de porta para as respectivas potências. Efetuando a aproximação polinomial de primeira ordem via Origin (equação 35), as tabelas 9, 10 e 11 mostram os dados extraídos de cada curva, sendo:

$$I_{CAT,int,linear}(V_G) = B_X . V_G + b_A \tag{35}$$

- b_A → Uma constante que intercepta I_{CAT},int(V_G);
- B_X → Parâmetro da equação (coeficiente angular);
- $I_{CAT,int,linear}(V_G)$ → Função da aproximação linear;

Figura 60 – Gráfico de $I_{CAT,int}(V_G)$ x V_G para faixa de V_G de 0V à 0,5V (experimental).



Fonte: Autor

É possível verificar nas tabelas que os coeficientes angulares (B_X) estão distintos entre si e o ERRO apresentado está em torno da ordem de grandeza de 10^{-14}, indicando 100% de aproximação.

Tabela 9 – Aproximação linear do azul (experimental).

PARÂMETROS	**0,2 W/m²**		**0,4 W/m²**		**0,6 W/m²**	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
b_A	3,13E-10	9,34E-14	4,36E-10	9,07E-14	5,66E-10	8,80E-14
B_X	1,94E-10	2,99E-13	3,73E-10	2,90E-13	5,60E-10	2,82E-13

Fonte: Autor

Tabela 10 – Aproximação linear do verde (experimental).

PARÂMETROS	**0,2 W/m²**		**0,4 W/m²**		**0,6 W/m²**	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
b_A	3,74E-10	9,22E-14	5,37E-10	8,88E-14	7,06E-10	8,62E-14
B_X	2,84E-10	2,95E-13	5,21E-10	2,84E-13	7,64E-10	2,76E-13

Fonte: Autor.

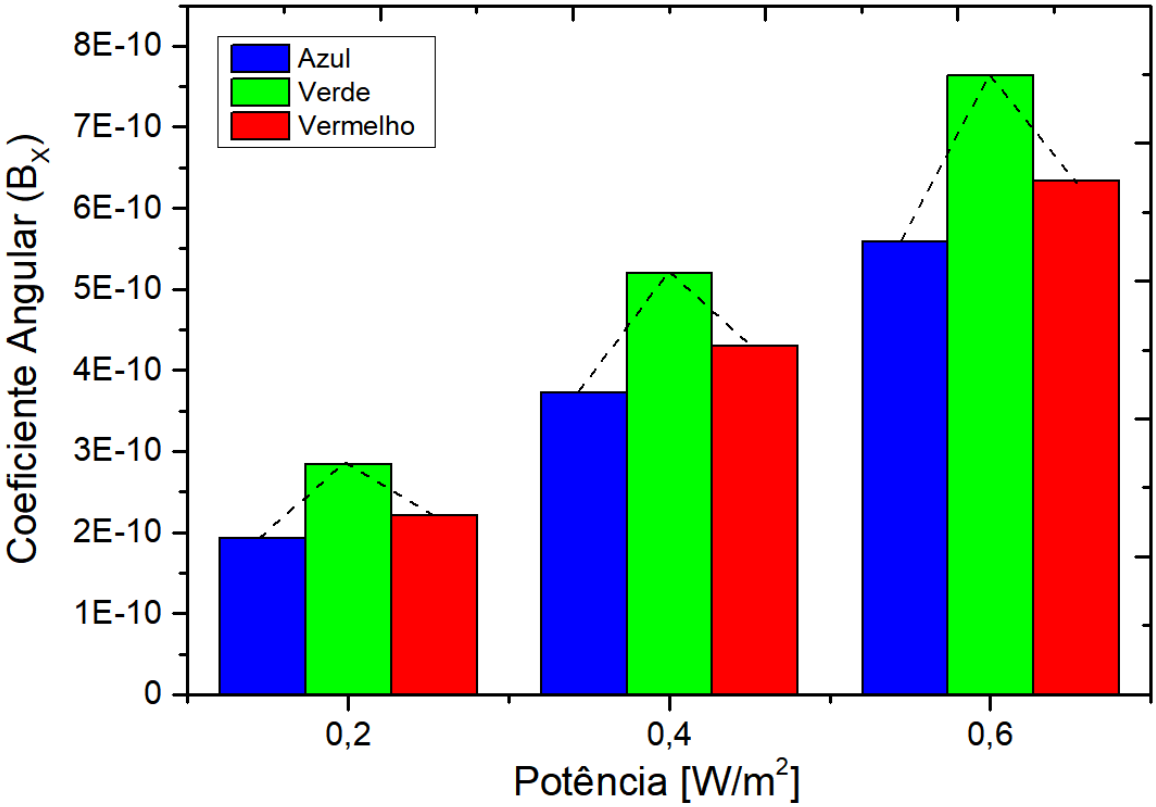
Tabela 11 – Aproximação linear do vermelho (experimental).

PARÂMETROS	**0,2 W/m²**		**0,4 W/m²**		**0,6 W/m²**	
	VALORES	ERRO	VALORES	ERROS	VALORES	ERROS
b_A	3,32E-10	9,29E-14	4,76E-10	9,03E-14	6,17E-10	8,74E-14
B_X	2,22E-10	2,97E-13	4,31E-10	2,89E-13	6,34E-10	2,80E-13

Fonte: Autor

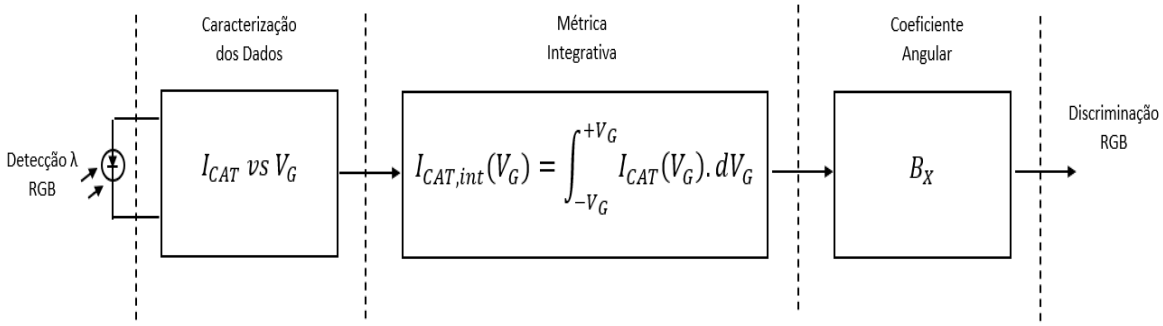
A fim de visualizar a discriminação das cores RGB através dos coeficientes angulares, a Tabela 11 mostra os coeficientes extraídos de cada curva e dispostos em pareto na Figura 61. Fica claro a distinção de cada cor, entre as cores para as potências de 0,2, 0,4 e 0,6 W/m^2, viabilizando assim o método integrativo. A Figura 62 mostra o diagrama passo a passo desse método.

Figura 61 – Gráfico de Coeficiente angular vs Potência do método integrativo (experimental).



Fonte: Autor

Figura 62 – Diagrama de blocos do método integrativo para discriminação RGB.

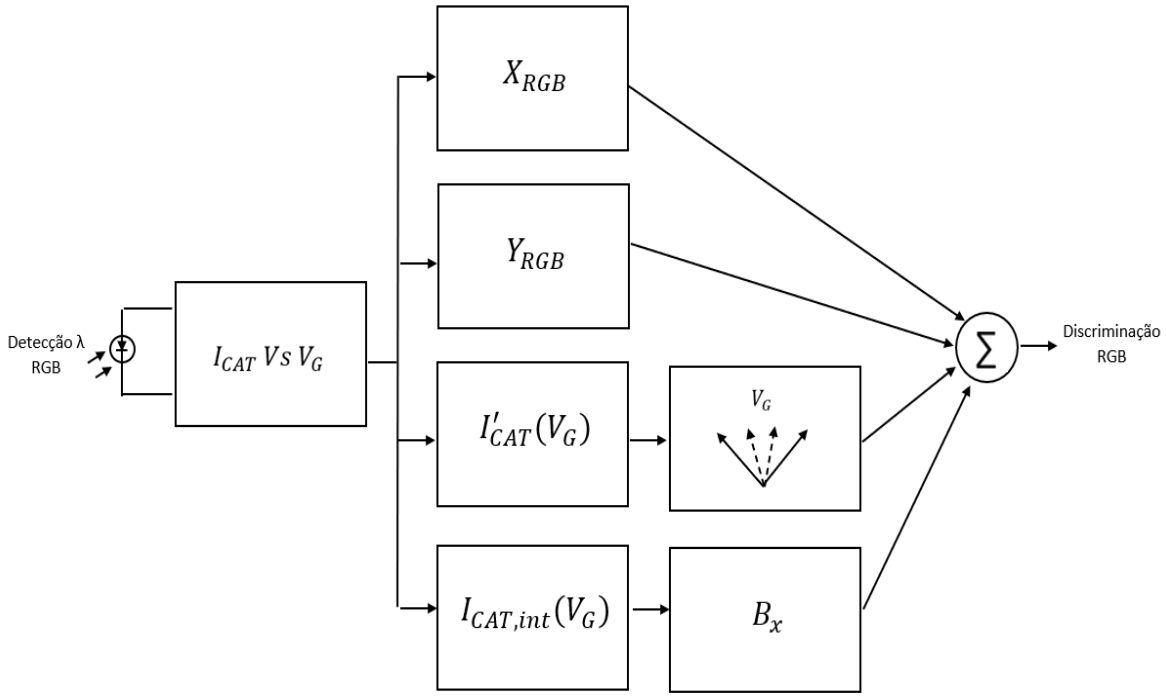


Fonte: Autor

6.4 RESULTADOS PARCIAIS DA CARACTERIZAÇÃO DO FOTODIODO PIN LATERAL COM PORTA

O capítulo 6 aborda toda a caracterização e discriminação do fotodiodo PIN lateral com porta baseado nos dados de I_{CAT} vs V_G. Inicialmente foi apresentado as curvas de funcionamento do dispositivo bem como algumas figuras de mérito, averiguando sua eficácia. Para a faixa de V_G entre -0,5V à +0,5V o máximo EQ_T alcançado foi para o comprimento de onda do verde com 39,98 %, seguido do vermelho com 33,23 % e do azul com 29,8 %. Já a máxima relação sinal ruído (SNR) ocorre para V_G = +0,46 V com o verde atingindo SNR = 44,35, seguido do vermelho com SNR = 43,54 e do azul com SNR = 42,99. Com intuito de entender e propor uma arquitetura para discriminar as cores RGB, foi necessário realizar um esforço massivo na mineração dos dados para encontrar métricas que norteia-se essa aplicação, sendo assim, é possível concluir que os índices X_{RGB} (razão das correntes de cada cor), Y_{RGB} (razão das corrente entre as cores), $I'_{CAT}(V_G)$ (método derivativo) e $I_{CAT,int}(V_G)$ (método integrativo), mostraram resultados extremamente promissores para a discriminação. A Figura 63 representa o compilado geral dos diagramas de blocos de cada método.

Figura 63 – Diagrama de blocos de todas as métricas para discriminação das cores RGB.



Fonte: Autor

7 RESULTADOS DOS MODELOS COMPUTACIONAIS GERADOS PELO APRENDIZADO DE MÁQUINA

Neste tópico serão tratados todos os resultados dos modelos computacionais gerados pelo aprendizado de máquina dos dados caracterizados e tratados do fotodiodo PIN lateral com porta discutido na secção 6. Na seção anterior foi constatado que o terminal de porta auxilia positivamente na discriminação das cores RGB, onde foi estudado e proposto 4 índices de mérito que medem a eficácia desta discriminação, sendo: X_{RGB}, Y_{RGB}, $I'_{CAT}(V_G)$ e $I_{CAT,int}(V_G)$.

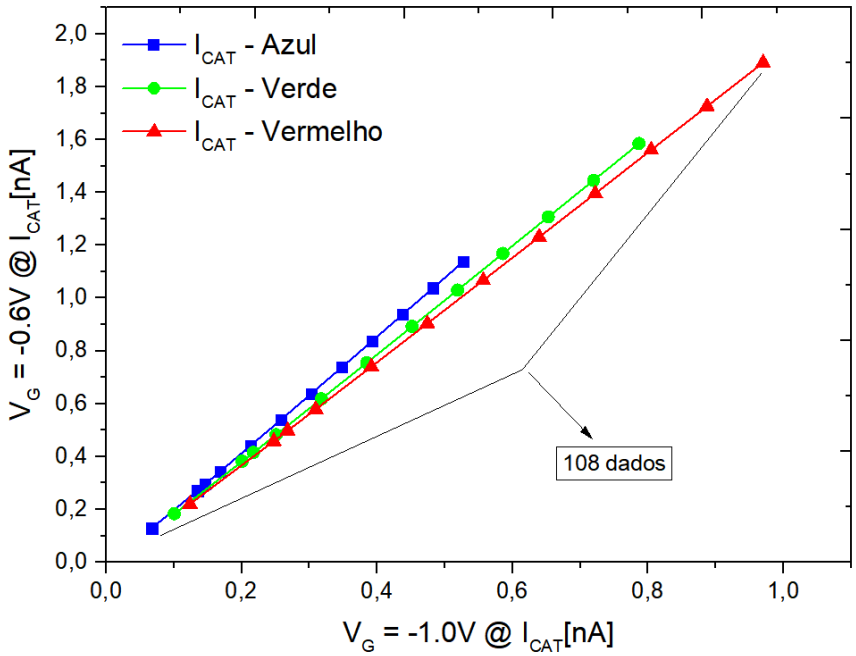
A quantidade total de dados estabelecidos para o aprendizado de máquina são 144, sendo que, 75% correspondem à 108 dados para treinamento e os 25% restantes (36 dados) para testar o modelo. A Tabela 12 indica o mapeamento dos dados que forão imputados no Matlab. Como exemplo de interpretação, toma-se a linha 1, que representa um vetor contendo a classe AZUL com os atributos de V_G = -1,0 V @ I_{CAT} = 6,74x10^{-10}A; V_G = -0,6 V @ I_{CAT} = 1,29x10^{-10}A; V_G = +0,6 V @ I_{CAT} = 1,85x10^{-10}; V_G = +0,60 V @ I_{CAT} = 1,85x10^{-10}; e a sim sucessivamente. As variáveis de entrada são os quatro valores de tensão de porta e a saída são as classes. Abaixo de cada variável V_G, estão dispostas todas as correntes para as diferentes potências ópticas incidentes da caracterização elétrica e das simulações discutidas no capítulo 6. Após imputados os dados, o próximo passo é definir a melhor dispersão para que o software inicie o treinamento. Segundo a Figura 64 a melhor dispersão foi a V_G = -0,6 V pelo V_G = -1,0 V.

Tabela 12 – Mapeamento dos dados I_{CAT} x V_G para faixa de potência de 0,1 à 1,0 W/m^2.

V_G = -1,0 V	**V_G = -0,6 V**	**V_G = +1,0 V**	**V_G = +0,6 V**	**CLASSE**
6.74x10^{-11} [A]	1.29x10^{-10} [A]	1.85x10^{-10} [A]	1.85x10^{-10} [A]	AZUL
1.35x10^{-10} [A]	2.71x10^{-10} [A]	3.68x10^{-10} [A]	3.68x10^{-10} [A]	AZUL
1.46x10^{-10} [A]	2.96x10^{-10} [A]	3.98x10^{-10} [A]	3.99x10^{-10} [A]	VERDE
1.00x10^{-10} [A]	1.83x10^{-10} [A]	2.57x10^{-10} [A]	2.57x10^{-10} [A]	VERDE
2.01x10^{-10} [A]	3.82x10^{-10} [A]	5.13x10^{-10} [A]	5.13x10^{-10} [A]	VERMELHO
2.47x10^{-10} [A]	4.58x10^{-10} [A]	6.33x10^{-10} [A]	6.34x10^{-10} [A]	VERMELHO
2.68x10^{-10} [A]	4.98x10^{-10} [A]	6.86x10^{-10} [A]	6.86x10^{-10} [A]	VERDE
...	...	...	...	...
TOTAL DE 108 DADOS PARA TREINAMENTO				

Fonte: Autor

Figura 64 – Gráfico dos dados V_G = -0,6V pelo V_G = -1,0 V para faixa de potência de 0,1 a 1,0 W/m^2 (Experimental + Simulado).



Fonte: Autor

Durante a fase de treinamento, o software propôs sete algoritmos com acurácia acima de 90% de acordo com a Tabela 13 que mostra o resultado de todas as acurácias dos 22 algoritmos de aprendizado de máquina.

Tabela 13 – Resultados dos algoritmos durante a fase de treinamento (experimental + simulado).

MODELOS	ACURÁCIA
Complex Tree	35,2 %
Medium Tree	35,2 %
Simple Tree	44,4 %
Linear Discriminant	**98,1 %**
Quadratic Discriminant	**100 %**
Linear SVM	**92,6 %**
Quadratic SVM	**98,1 %**
Cubic SVM	**96,3 %**
Fine Gaussian SVM	40,7 %
Medium Gaussian SVM	46,3 %

MODELOS	ACURÁCIA
Coarse Gaussian SVM	44,4 %
Fine KNN	36,1 %
Medium KNN	43,5 %
Coarse KNN	32,4 %
Cosine KNN	**91,7 %**
Cubic KNN	43,5 %
Weighted KNN	39,8 %
Ensemble Boosted Trees	33,3 %
Ensemble Bagged Trees	29,6 %
Ensemble Subspace Discriminant	**90,7 %**
Ensemble Subspace KNN	21,3 %
Ensemble RUSBossted Trees	38,9 %

Fonte: Autor

Os modelos a serem analisados são *Linear Discriminant*, *Quadratic Discriminant*, *Linear SVM*, *Quadratic SVM*, *Cubic SVM*, *Cosine KNN* e *Ensemble Subspace Discriminant*. Uma vez com os modelos definidos, os próximos passos são determinar as métricas discutidas no capítulo 4. A matriz de confusão dos sete algoritmos é apresentada nas Tabelas 14 a 20. A interpretação dessa matriz ocorre da seguinte maneira: tomemos como exemplo a matriz da Tabela 14, a quantidade total de dados para o treinamento é 108, sendo 36 dados para cada classe. Para os 36 dados da classe azul, o algoritmo classificou corretamente todos como azul, dos 36 dados da classe verde, o algoritmo classificou corretamente 35 dados como verde e um dado errado como azul e por fim, dos 36 dados da classe vermelho, o algoritmo classificou corretamente 35 dados como vermelho e um dado errado como azul.

Tabela 14 – Matriz de confusão do algoritmo *Linear Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	36	0	0
	VERDE	1	35	0
	VERMELHO	1	0	35

Fonte: Autor

Tabela 15 – Matriz de confusão do algoritmo *Quadratic Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	36	0	0
	VERDE	0	36	0
	VERMELHO	0	0	36

Fonte: Autor

Tabela 16 – Matriz de confusão do algoritmo *Linear SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	36	0	0
	VERDE	2	34	0
	VERMELHO	1	5	30

Fonte: Autor

Tabela 17 – Matriz de confusão do algoritmo *Quadratic SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	36	0	0
	VERDE	0	36	0
	VERMELHO	0	2	34

Fonte: Autor

Tabela 18 – Matriz de confusão do algoritmo *Cubic SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	36	0	0
	VERDE	0	35	1
	VERMELHO	0	3	33

Fonte: Autor

Tabela 19 – Matriz de confusão do algoritmo *Cosine KNN*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	33	3	0
	VERDE	0	36	0
	VERMELHO	0	6	30

Fonte: Autor

Tabela 20 – Matriz de confusão do algoritmo *Ensemble Subspace Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	32	4	0
	VERDE	0	36	0
	VERMELHO	0	6	30

Fonte: Autor

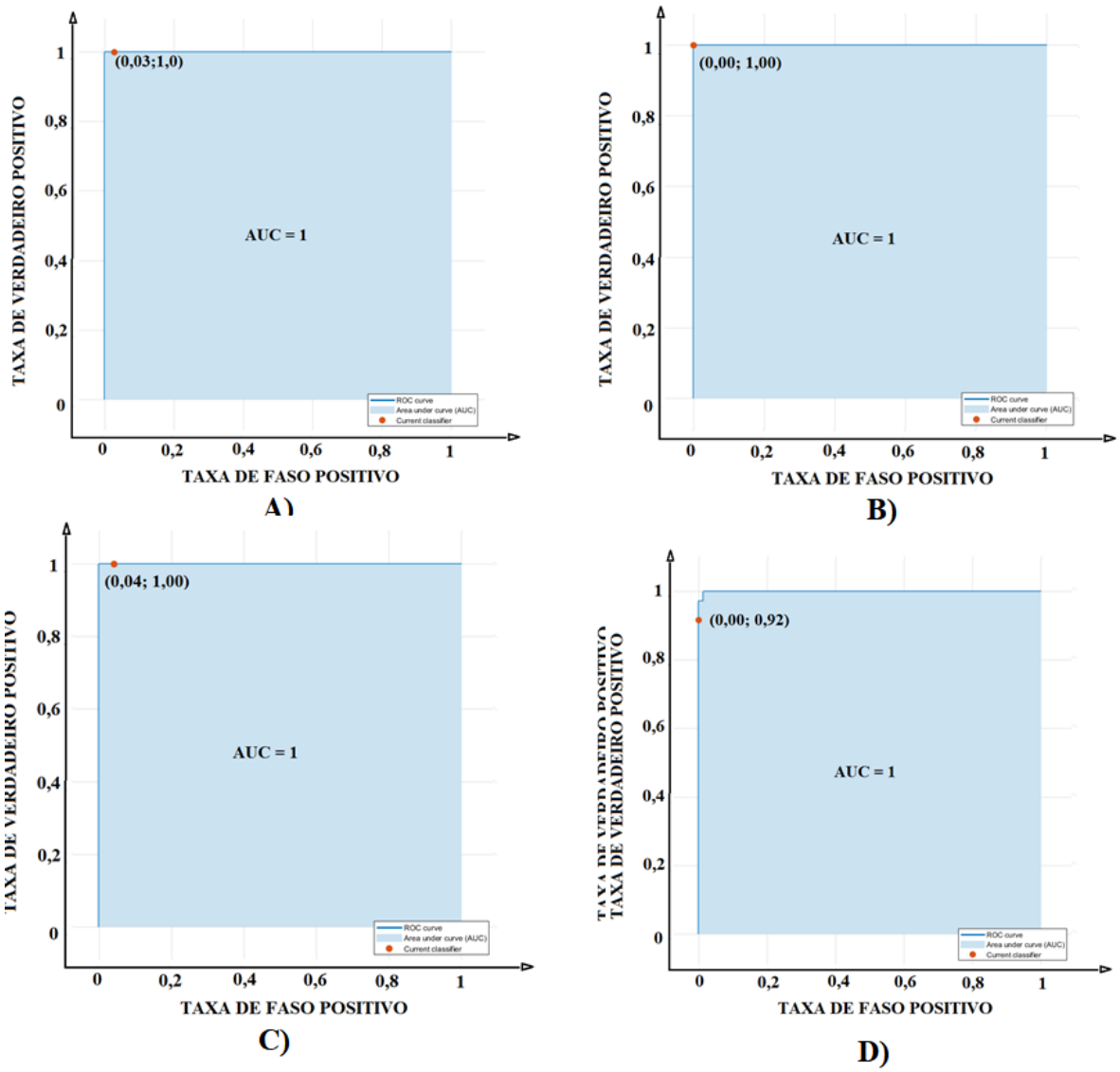
Além das matrizes de confusão, as Figuras 65-A) *Linear Discriminant*, 65-B) *Quadratic Discriminant, Quadratic SVM* e *Cubic SVM*, 65-C) Linear SVM, 65-D) *Cosine KNN* e 65 E-) *Ensemble Subspace Discriminant* mostram as curvas ROC/AUC que medem o grau de separabilidade entre as classes. Esse índice varia de 0 a 1, quanto mais próximo de 1, significa que o algoritmo consegue distinguir com eficiência as classes do azul, verde e vermelho. A Tabela 21 mostra o resumo do AUC/ROC de cada algoritmo.

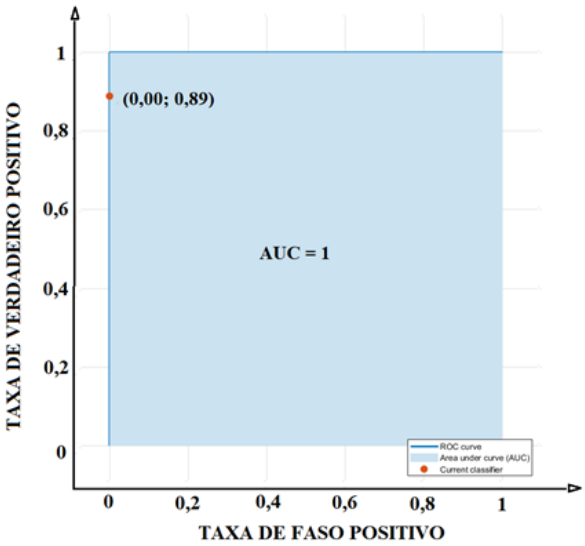
Tabela 21 – Resumo do resultado ROC/AUC de cada algoritmo.

ALGORITMO	ROC/AUC	TPR	FPR
Linear Discriminant	1,00	1,00	0,03
Quadratic Discriminant	1,00	1,00	0,00
Linear SVM	1,00	1,00	0,04
Quadratic SVM	1,00	1,00	0,00
Cubic SVM	1,00	1,00	0,00
Cosine KNN	1,00	0,92	0,00
Ensemble Subspace Discriminant	1,00	0,89	0,00

Fonte: Autor

Figura 65 – Gráficos ROC/ACU.





E)

Fonte: Autor

A Tabela 22 mostra o resumo da fase de treinamento.

Tabela 22 – Resumo dos algoritmos durante a fase de treinamento.

RESULTADOS	***Linear Discriminant***	***Quadratic Discriminant***	***Linear SVM***
Qt. Dados	108	108	108
Acurácia	98,10 %	100%	92,60%
Precision	0,9815	1,0	0,9259
Recall	0,9825	1,0	0,9316
F-Score	0,9820	1,0	0,9288
ROC/AUC	1,0	1,0	1,0
Classes Corretas (B)	36	36	36
Classes Corretas (G)	35	36	34
Classes Corretas (R)	35	36	30
Classes Erradas	2	0	8
Velocidade de Predição	360 obs./seg.	850 obs./sec.	540 obs./sec.
Tempo de Treinamento	11,192 seg.	16,495 sec.	15,771 sec.

Quadratic SVM	*Cubic SVM*	*Cosine KNN*	*Ensemble Subspace Discriminant*
108	108	108	108
98,10%	96,30%	91,7 %	90,7 %
0,9815	0,9630	0,9167	0,9074
0,9825	0,9639	0,9333	0,9275
0,9820	0,9634	0,9249	0,9174
1,0	1,0	1,0	1,0
36	36	33	32
36	35	36	36
34	33	30	30
2	4	9	10
620 obs./sec.	530 obs./sec.	970 obs./sec.	130 obs./sec.
45,218 sec.	15,175 sec.	17,044 sec.	23,199 sec.

Fonte: Autor

O próximo passo é inicializar a fase de teste para testar de fato o quão eficiente são os algoritmos em classificar as cores RGB. Nesta etapa, o algoritmo tem que receber um arquivo que contenha os novos dados do mesmo formato que foi imputado na fase de treinamento. Sendo, quatro colunas de V_G = -1 V, -0,6 V, +1 V, +0,6 V e uma coluna CLASSE, onde nessa coluna o algoritmo irá classificar as cores RGB, indicado pelo círculo vermelho. A Tabela 23 mostra o formato do arquivo.

Tabela 23 – Novos dado imputados para testar os algoritmos.

V_G = -1,0 V	**V_G = -0,6 V**	**V_G = +1,0 V**	**V_G = +0,6 V**	**CLASSE**
6.74×10^{-11} [A]	1.29×10^{-10} [A]	1.85×10^{-10} [A]	1.85×10^{-10} [A]	RGB ?
1.35×10^{-10} [A]	2.71×10^{-10} [A]	3.68×10^{-10} [A]	3.68×10^{-10} [A]	
1.46×10^{-10} [A]	2.96×10^{-10} [A]	3.98×10^{-10} [A]	3.99×10^{-10} [A]	
...	...	...	...	
TOTAL DE 36 DADOS PARA FASE DE TESTE				

Fonte: Autor

Após efetuado a classificação, foi desenvolvido um *script* no MATLAB descrito no apêndice A para calcular a acurácia e gerar o resultado da matriz de confusão. As Tabelas 24 a 30 mostram os resultados das matrizes de confusão de cada algoritmo durante a fase de teste e a tabela 31 mostra o resumo completo dos demais índices. A capacidade de classificação das cores RGB dos algoritmos mostram-se positivas, com acurácia acima de 94%, sendo que três algoritmos, *Linear Discriminant*, *Quadratic Discriminant* e o *Quadratic SVM* tiveram 100 % de acurácia.

Tabela 24 – Matriz de confusão do algoritmo *Linear Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	0	12	0
	VERMELHO	0	0	12

Fonte: Autor

Tabela 25 – Matriz de confusão do algoritmo *Quadratic Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	0	12	0
	VERMELHO	0	0	12

Fonte: Autor

Tabela 26 – Matriz de confusão do algoritmo *Linear SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	1	11	0
	VERMELHO	0	1	11

Fonte: Autor

Tabela 27 – Matriz de confusão do algoritmo *Quadratic SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	0	12	0
	VERMELHO	0	0	12

Fonte: Autor

Tabela 28 – Matriz de confusão do algoritmo *Cubic SVM*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	0	12	0
	VERMELHO	0	1	11

Fonte: Autor

Tabela 29 – Matriz de confusão do algoritmo *Cosine KNN*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	12	0	0
	VERDE	0	12	0
	VERMELHO	0	1	12

Fonte: Autor

Tabela 30 – Matriz de confusão do algoritmo *Ensemble Subspace Discriminant*.

		VALORES PREDITOS		
		AZUL	VERDE	VERMELHO
VALORES REAIS	AZUL	11	1	0
	VERDE	0	12	0
	VERMELHO	0	1	11

Fonte: Autor

Tabela 31 – Resumo dos algoritmos durante a fase de teste.

RESULTADOS	***Linear Discriminant***	***Quadratic Discriminant***	***Linear SVM***	***Quadratic SVM***	***Cubic SVM***	***Cosine KNN***	***Ensemble Subspace Discriminant***
Qt. Dados	36	36	36	36	36	36	36
Acurácia	100 %	100 %	94,44 %	100 %	97,22 %	97,22 %	94,44 %
Precision	1,00	1,00	0,9444	1,00	0,9722	0,9722	0,9444
Recall	1,00	1,00	0,9446	1,00	0,9744	0,9744	0,9524
F-Score	1,00	1,00	0,9455	1,00	0,9733	0,9733	0,9484
Classes Corretas (B)	12	12	12	12	12	12	11
Classes Corretas (G)	12	12	11	12	12	12	12
Classes Corretas (R)	12	12	11	12	11	11	11
Classes Erradas	0	0	2	0	1	1	2

Fonte: Autor.

Após realizado a primeira rodada de classificação, é importante fazer mais algumas rodadas para averiguar a média e a dispersão dos modelos (A MOSAVI P OZTURK AND K CHAU, [s.d.]). Lembrado que cada rodada é composta da fase de treinamento (108 dados) + fase de teste (36 dados). A escolha dos dados de cada fase foi escolhida de forma aleatória para não enviesar os resultados de cada rodada, ou seja, a base principal de dados é a mesma para

todos os modelos, composta de 144 dados, os 75% correspondem 108 dados escolhidos de forma aleatória e da mesma forma para os 25 % que correspondem 36 dados. A Tabela 32 e 33 mostra todas as rodadas com as respectivas acurácias, médias e o desvio padrão de cada modelo. Para a fase de treinamento a média da acurácia para todos os modelos ficaram acima dos 90%, já para a fase de teste, todos os modelos ficaram acima de 80%.

Tabela 32 – Acurácia, média e desvio padrão para 7 rodadas da fase de treinamento.

FASE TREINAMENTO	***Linear Discriminant***	***Quadratic Discriminant***	***Linear SVM***
RODADA 1 (ACURÁCIA)	98,1%	100%	92,6%
RODADA 2 (ACURÁCIA)	97,2%	98,1%	94,4%
RODADA 3 (ACURÁCIA)	95,3%	100%	88,7%
RODADA 4 (ACURÁCIA)	95,3%	100%	91,5%
RODADA 5 (ACURÁCIA)	97,2%	100%	88,7%
RODADA 6 (ACURÁCIA)	99,1%	100%	87,7%
RODADA 7 (ACURÁCIA)	97,2%	100%	88,7%
MÉDIA	**97,06%**	**99,73%**	**90,33%**
DESVIO PADRÃO	**1,38%**	**0,72%**	**2,51%%**

FASE TREINAMENTO	***Quadratic SVM***	***Cubic SVM***	***Cosine KNN***	***Ensemble Subspace Discriminant***
RODADA 1 (ACURÁCIA)	98,1%	96,3%	96,3%	90,7%
RODADA 2 (ACURÁCIA)	96,3%	95,3%	90,7%	98,1%
RODADA 3 (ACURÁCIA)	94,3%	86,8%	92,5%	96,2%
RODADA 4 (ACURÁCIA)	94,3%	87,7%	88,7%	90,6%
RODADA 5 (ACURÁCIA)	97,2%	90,6%	93,4%	94,3%
RODADA 6 (ACURÁCIA)	89,6%	91,5%	91,5%	86,8%
RODADA 7 (ACURÁCIA)	90,6%	95,3%	94,3%	87,7%
MÉDIA	**94,34%**	**91,93%**	**92,49%**	**92,06%**
DESVIO PADRÃO	**3,23%**	**3,82%**	**2,49%**	**4,27%**

Fonte: Autor

Tabela 33 – Acurácia, média e desvio padrão para 7 rodadas da fase de teste.

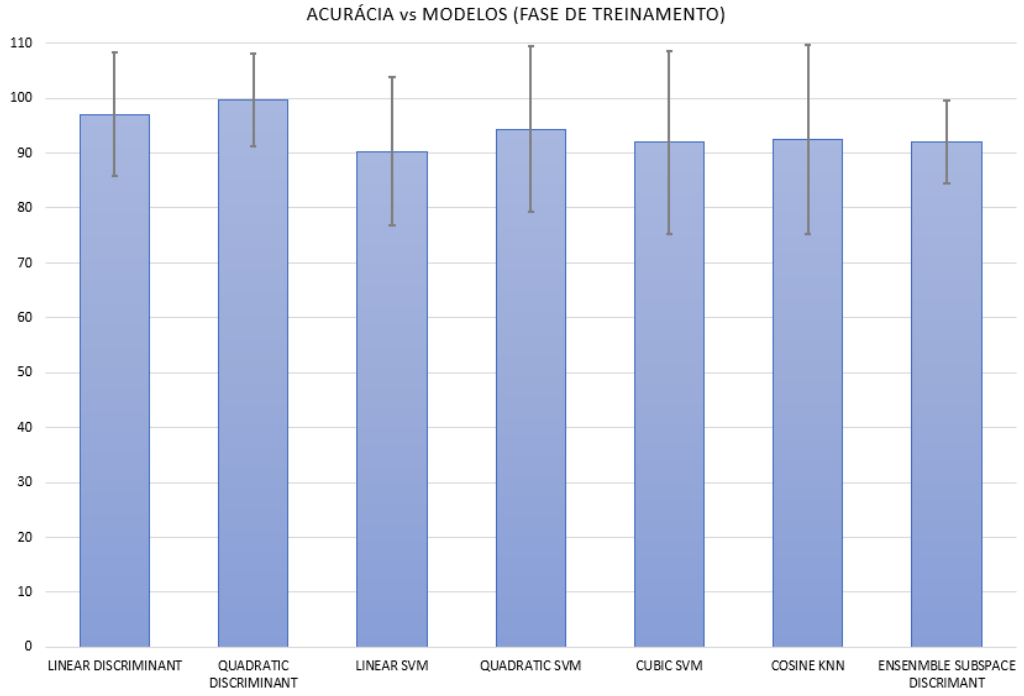
FASE TREINAMENTO	***Linear Discriminant***	***Quadratic Discriminant***	***Linear SVM***
RODADA 1 (ACURÁCIA)	100%	100%	94,44%
RODADA 2 (ACURÁCIA)	75%	100%	72,22%
RODADA 3 (ACURÁCIA)	94,44%	100%	72,22%
RODADA 4 (ACURÁCIA)	100%	100%	100%
RODADA 5 (ACURÁCIA)	77,78%	77,78%	61,11%
RODADA 6 (ACURÁCIA)	100%	100%	80,56%
RODADA 7 (ACURÁCIA)	100%	100%	83,33%
MÉDIA	**92,46%**	**96,82%**	**80,55%**
DESVIO PADRÃO	**11,19%**	**8,39%**	**13,51%**

FASE TREINAMENTO	***Quadratic SVM***	***Cubic SVM***	***Cosine KNN***	***Ensemble Subspace Discriminant***
RODADA 1 (ACURÁCIA)	100%	97,22%	97,22%	94,44%
RODADA 2 (ACURÁCIA)	69,44%	88,89%	61,11%	100%
RODADA 3 (ACURÁCIA)	86,11%	66,67%	55,56%	80,56%
RODADA 4 (ACURÁCIA)	100%	100%	83,33%	100%
RODADA 5 (ACURÁCIA)	66,67%	65,3%	69,44%	100%
RODADA 6 (ACURÁCIA)	100%	58,33%	94,44%	88,89%
RODADA 7 (ACURÁCIA)	100%	75%	94,44%	100%
MÉDIA	**88,88%**	**78,77%**	**79,36%**	**94,84%**
DESVIO PADRÃO	**15,13%**	**16,60%**	**17,26%**	**7,59%**

Fonte: Autor

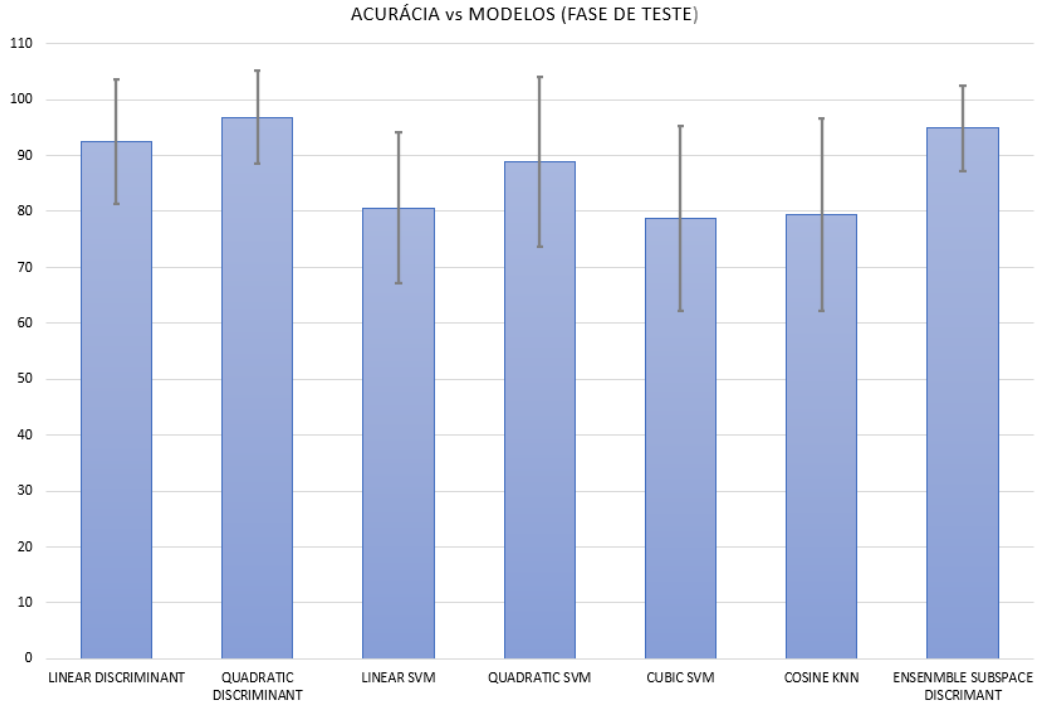
Na fase de treinamento os modelos que obtiverem maiores e menores médias das acurácias respectivamente foram *Quadratic SVM* (99,73%) e o *Linear SVM* (90,32%). Já na fase de teste os modelos que obtiveram maiores e menores médias das acurácias foram *Quadratic SVM* (96,83%) e o *Cubic SVM* (78,77%). As Figuras 66 e 67 mostram o resultado das médias e dos desvios padrões em gráfico de barras como comparativo entre os modelos.

Figura 66 – Quadro comparativo das médias das acurácias entre os modelos durante a fase de treinamento.



Fonte: Autor

Figura 67 – Quadro comparativo das médias das acurácias entre os modelos durante a fase de teste.

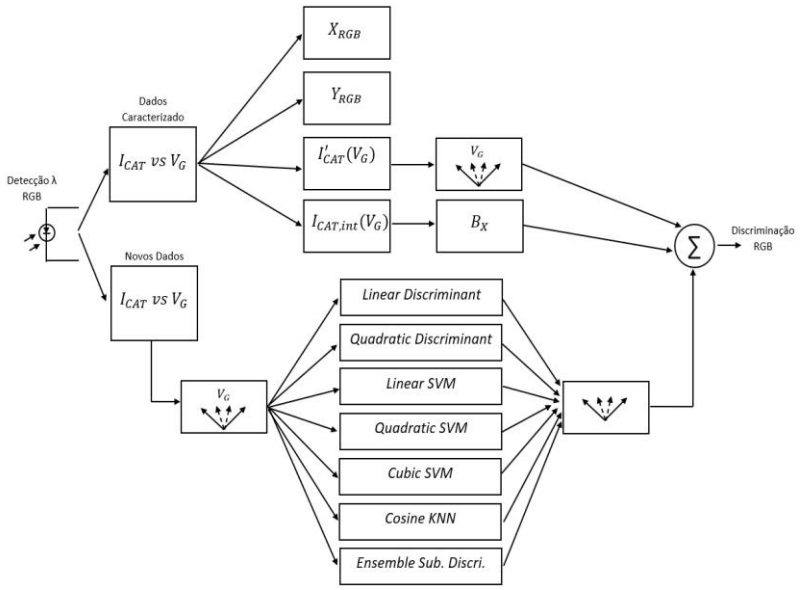


Fonte: Autor

7.4 RESULTADOS PARCIAIS DOS ALGORITMOS GERADOS PELO APRENDIZADO DE MÁQUINA

O capítulo 7 aborda a obtenção dos algoritmos do aprendizado de máquina para os dados caracterizados e simulados do fotodiodo PIN Lateral com porta. Dos 22 algoritmos disponibilizados, sete apenas tiveram acurácia acima de 90 % e ROC/AUC = 1 durante a fase de treinamento, na primeira rodada, sendo: *Linear Discriminant, Quadratic Discriminant, Linear SVM, Quadratic SVM, Cubic SVM, Cosine KNN* e o *Ensemble Subspace Discriminant*. Um erro comum de iniciantes no aprendizado de máquina é achar que, pelo fato das métricas dos algoritmos se mostrem satisfatórias na fase de treinamento, não significa que o mesmo é um bom classificador (A MOSAVI P OZTURK AND K CHAU, [s.d.]). Para se ter uma maior confiabilidade dos modelos, foram realizadas 7 rodas de treinamento/teste para averiguar a média e o desvio padrão dos modelos para a classificação dos dados, logo, na fase de treinamento os modelos que obtiverem maiores e menores médias das acurácias respectivamente foram *Quadratic SVM* (99,73%) e o *Linear SVM* (90,32%). Já na fase de teste os modelos que obtiveram maiores e menores médias das acurácias foram *Quadratic SVM* (96,83%) e o *Cubic SVM* (78,77%). A Figura 68 representa o compilado geral dos diagramas de blocos dos sete algoritmos do aprendizado de máquina juntamente com as métricas discutidas no capítulo 6.

Figura 68 – Diagrama de blocos dos sete algoritmos do aprendizado de máquina para classificar as cores RGB.

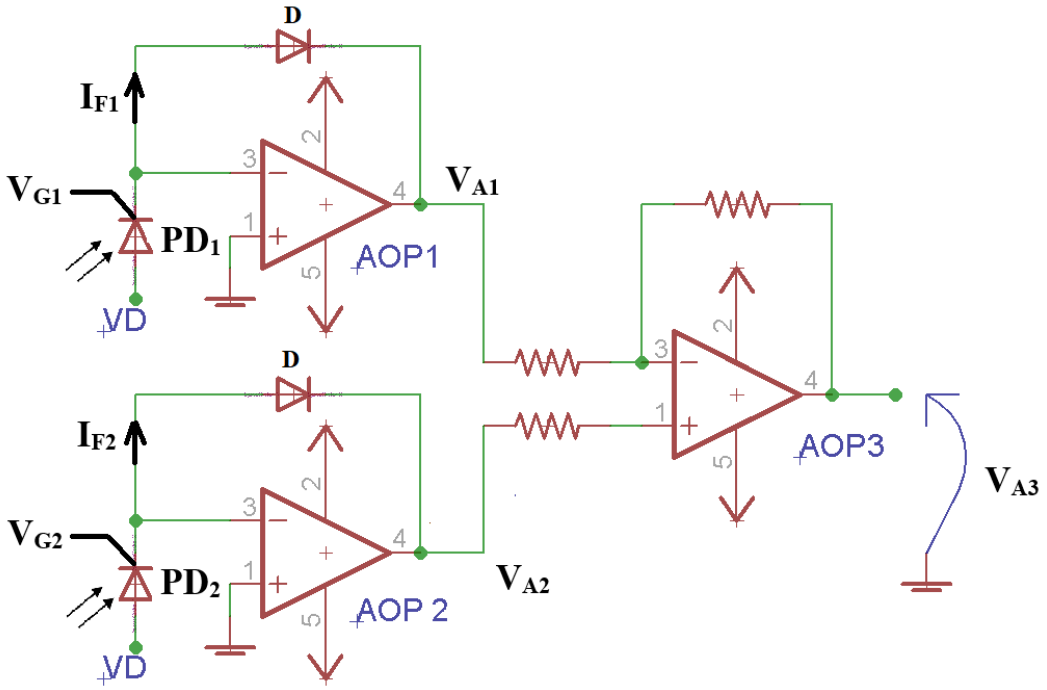


Fonte: Autor

8 CIRCUITO ELETRÔNICO DE DISCRIMINAÇÃO DAS CORES RGB

Nos capítulos anteriores foram discutidas as discriminações das cores RGB através dos índices de méritos X_{RGB}, Y_{RGB}, $I'_{CAT}(V_G)$, $I_{CAT,int}(V_G)$ e dos algoritmos computacionais gerados pelo AM. Mediante a estes resultados, este capítulo irá propor um circuito eletrônico baseado no circuito discutido no item 3.6 (ASO; KAWANABE, 1982), que tem como objetivo encontrar uma relação entre a tensão elétrica de saída com o comprimento de onda. Porém, esse circuito trabalha com um fotodiodo vertical constituído de duas regiões PN para detectar os comprimentos de ondas curtos e longos. A alteração nesse circuito é a utilização de dois fotodiodos idênticos PIN lateral com porta, onde as faixas de V_G discutidas nos capítulos 6 e 7 detectarão os comprimentos curtos e longos, conforme Figura 69.

Figura 69 – Circuito eletrônico para detecção dos comprimentos de onda.



Fonte: Autor

A configuração dos amplificadores operacionais AOP1 e AOP2 é conhecida como logarítmica (BAKER, 1997; RAVAZI, 2010) devido a realimentação usar um diodo convencional PN. Essa configuração traz consigo a relação entre a tensão de saída V_{A1} e V_{A2} com as correntes de entrada I_{F1} e I_{F2}, que são as correntes dos fotodiodos PD_1 e PD_2. As Equações (36) e (37) são as respectivas tensões de saída V_{A1} e V_{A2}, sendo I_S a corrente de

saturação do diodo D e V_K a tensão térmica (K.T/q) que em temperatura ambiente (25º C) é aproximadamente 25mV (BAKER, 1997) .

$$V_{A1} = -V_K . ln\left(\frac{I_{F1}}{I_S}\right) \tag{36}$$

$$V_{A2} = -V_K . ln\left(\frac{I_{F2}}{I_S}\right) \tag{37}$$

O AOP3 é a configuração do amplificador de diferenças, onde a tensão $V_\lambda = V_{A2} - V_{A1}$. Substituindo as Equações (36) e (37) na tensão V_λ, o resultado é mostrado na Equação (38), ou seja, a tensão V_λ é a razão logarítmica entre as correntes I_{F1} pela I_{F2}. Essas correntes são selecionadas pelas tensões de portas V_{G1} e V_{G2}.

$$V_\lambda = -V_K . ln\left(\frac{I_{F2}}{I_S}\right) + V_K . ln\left(\frac{I_{F1}}{I_S}\right) = V_K . ln\left[\frac{\frac{I_{F1}}{I_S}}{\frac{I_{F2}}{I_S}}\right] \rightarrow V_\lambda = V_K . ln\left(\frac{I_{F1}}{I_{F2}}\right) \tag{38}$$

As tensões V_{G1} e V_{G2} que serão utilizadas na equação (38) são as mesmas propostas nos algoritmos do AM do capítulo 7. A Figura 70 mostra exatamente o benefício da tensão de porta no auxílio da detecção dos comprimentos de onda. Essa análise é feita com a potência óptica de 0,5 W/m^2 e V_G = +0,6 V, -0,6 V, -1 V e +1V. Nota-se que à medida que a tensão de porta aumenta a corrente de catodo aumenta, demostrando a importância de V_G. Esse comportamento ocorre, devido a polarização V_G contribui positivamente na RDD vertical do dispositivo, acarretando numa maior taxa na geração de portadores. Entretanto quando V_G > +0,6 V, segundo a Figura 70, essa contribuição não é tão pronunciada, dando indícios de que, o dispositivo já está na taxa máxima de geração de portadores dentro da RDD. Mediante a isso, os valores de V_G usados na equação, serão: -1 V, -0,6 V e +1,0 V. Como o circuito eletrônico necessita de dois valores de V_{G1} e V_{G2}, é possível gerar três combinações destes valores, segundo a Tabela 34. As Figuras 71 a 73 mostram o resultado da equação (38) para as potências de 0,1; 0,5 e 1 W/m^2. Nota-se que para a potência de 0,1 W/m^2 as três relações entre os V_G têm boa discriminação entre os comprimentos de onda entre 200 nm à 500 nm. Acima de 500 nm a variação de V_λ é praticamente imperceptível, pois nessa faixa a resposta espectral tende a ser constante, conforme Figura 71. Com o aumento das potências (Figuras 71 a 73), a relação entre

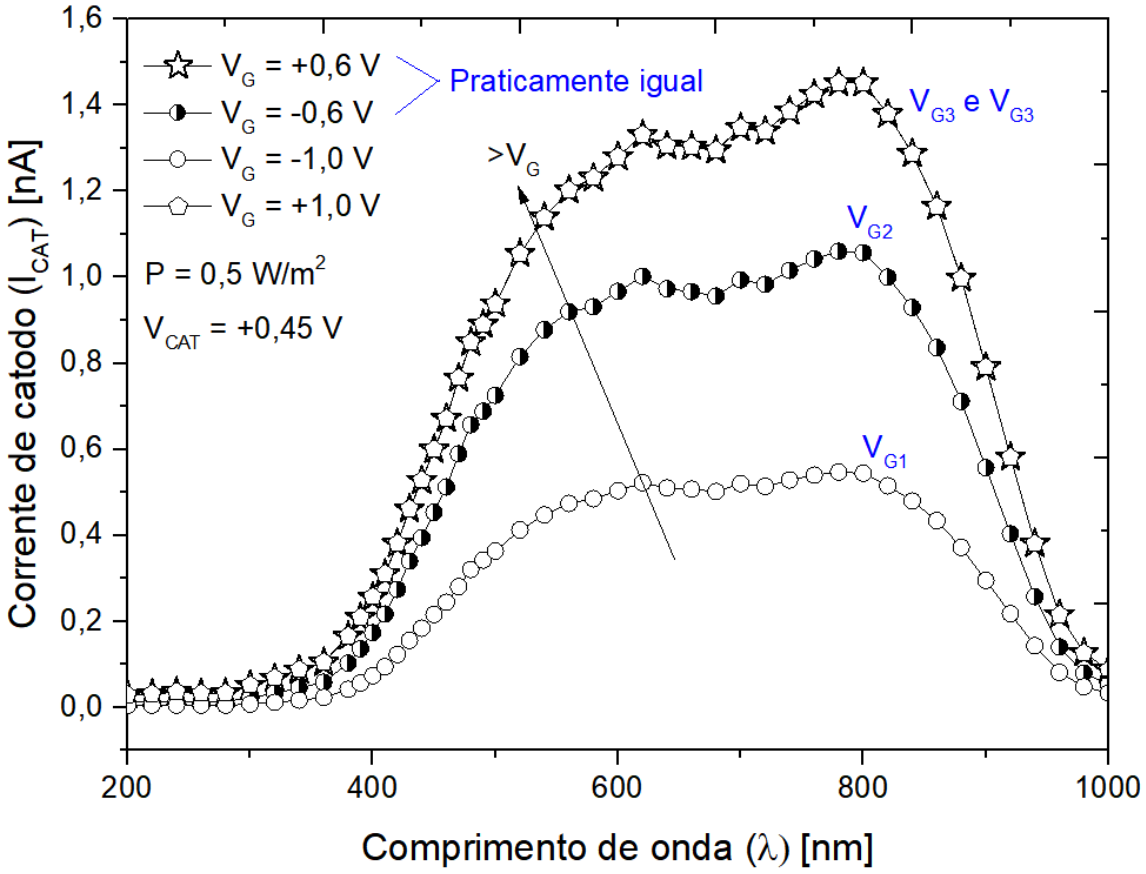
os IF_{VG1} = + 1,0 V/ IF_{VG2} = - 0,6 V em toda a faixa de 200 nm à 100 nm possui a melhor relação entre V_λ por λ. A faixa de tensão V_λ está compreendia entre 0 e 50 mV. As Figuras 74 a 76 mostram o comparativo dos resultados entre as potências analisadas nas Figuras 71 a 73.

Tabela 34 – Combinação dos valores de V_{G1} e V_{G2}.

V_{G1}	V_{G2}
-0,6 V	- 1 V
+1 V	-1 V
+1 V	-0,6 V

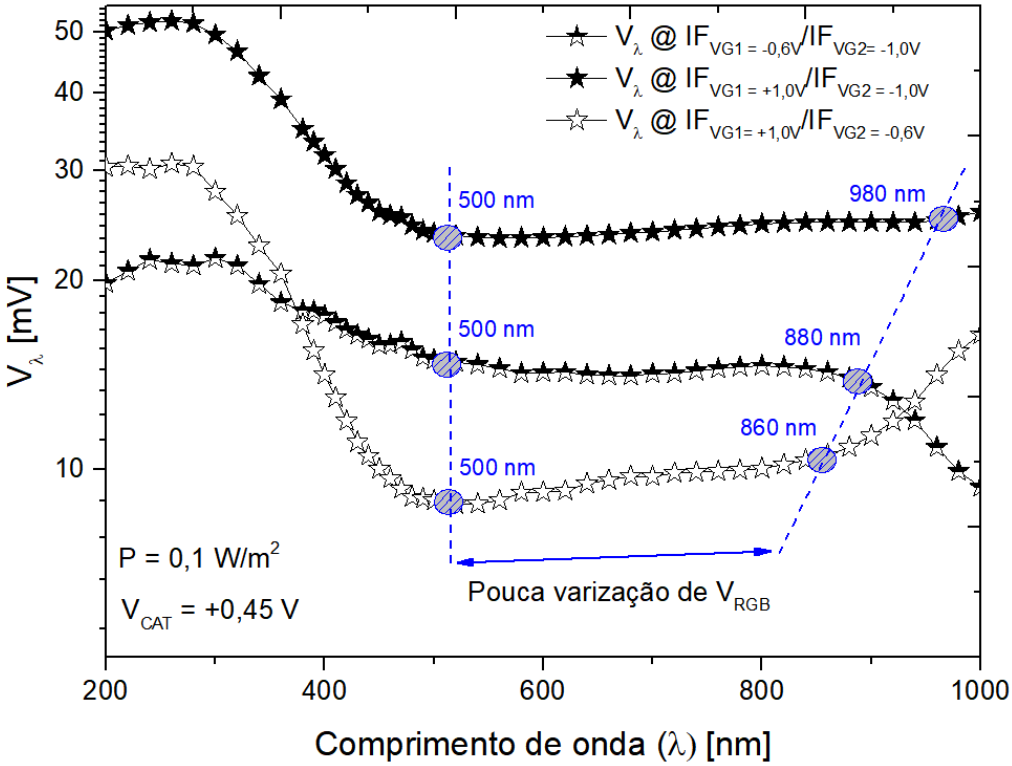
Fonte: Autor

Figura 70 – I_{CAT} vs λ com P = 0,5 W/m^2 para as quatro tensões V_G usados nos algoritmos do AM.



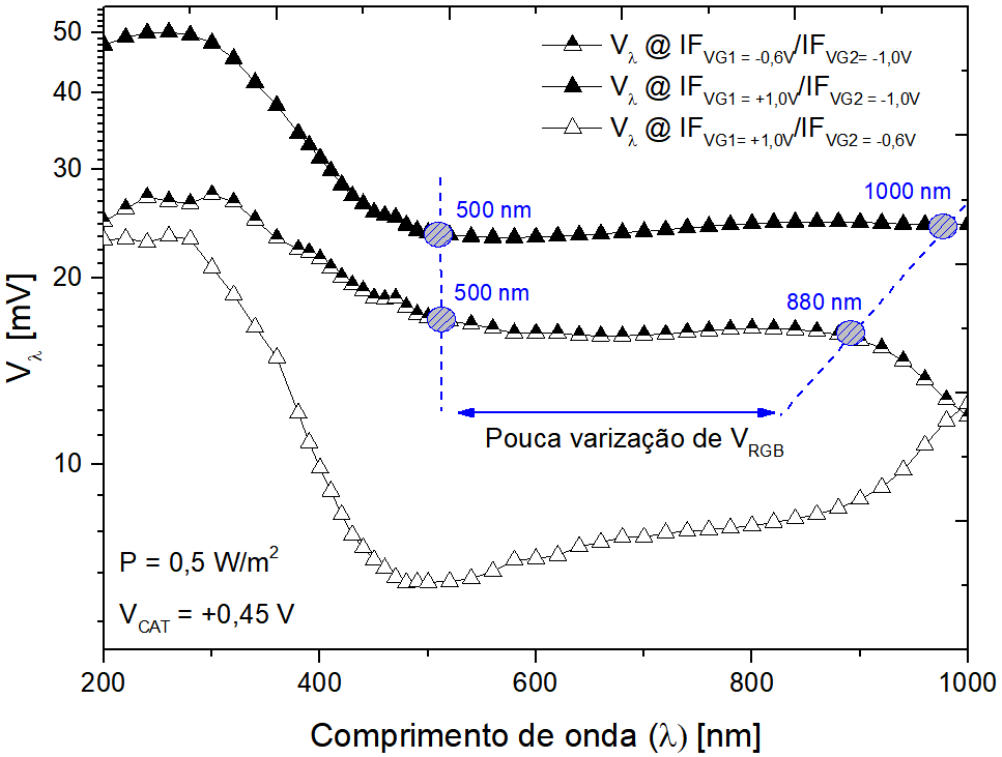
Fonte: Autor

Figura 71 – V_λ vs λ com P = 0,1 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V.



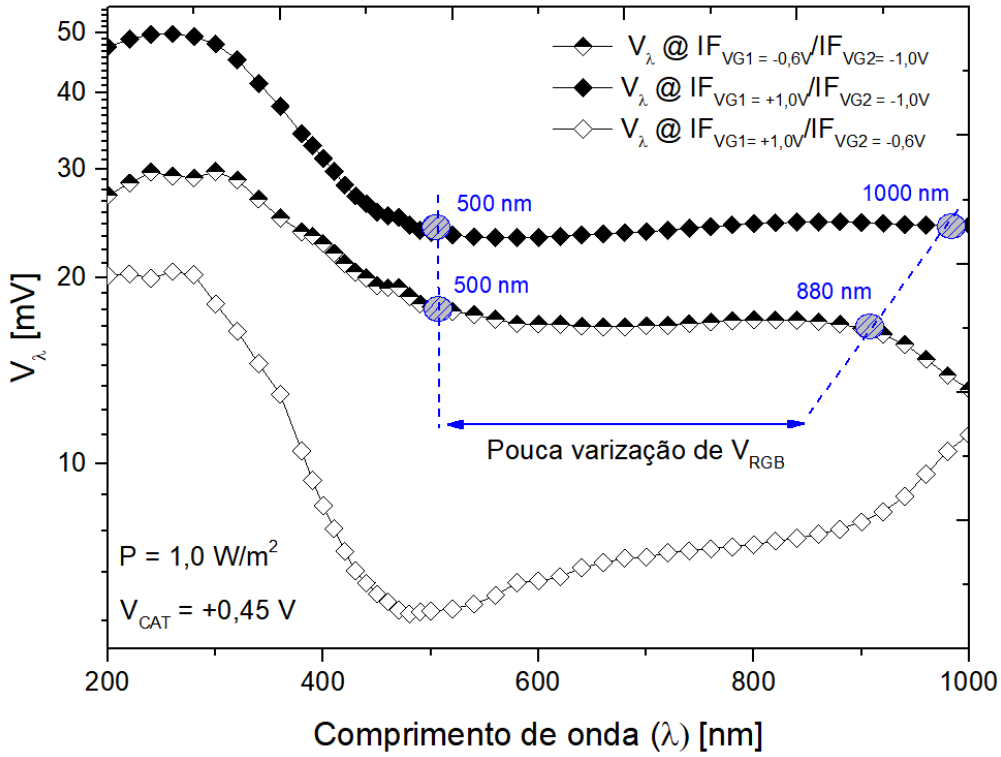
Fonte: Autor

Figura 72 – V_λ vs λ com P = 0,5 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V.



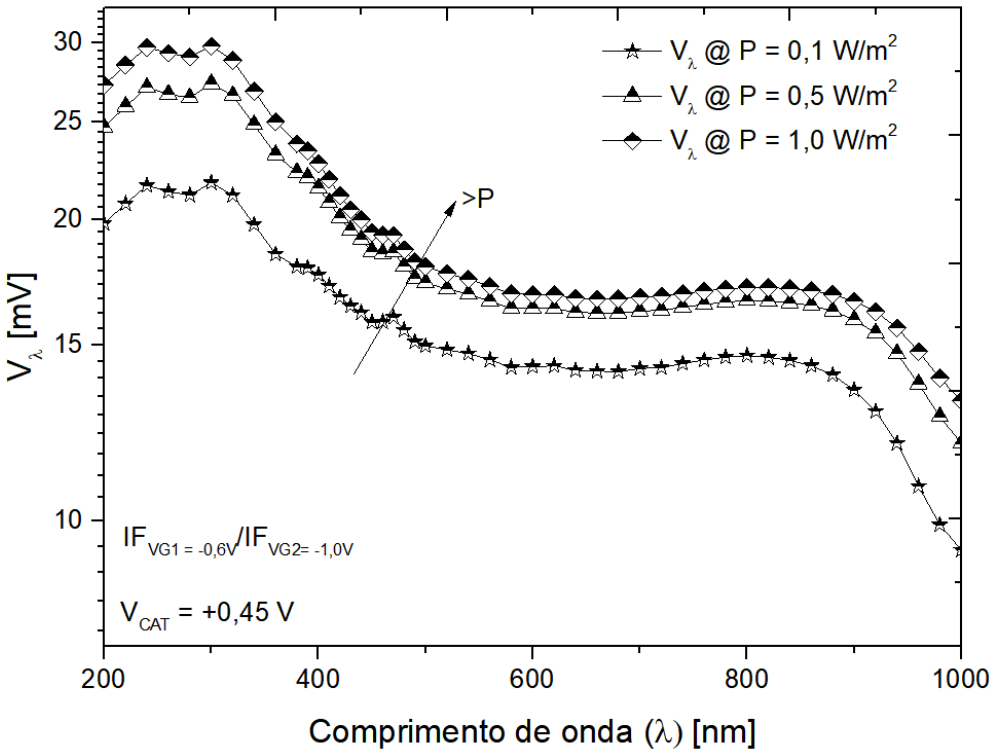
Fonte: Autor

Figura 73 – V_λ vs λ com P = 1,0 W/m^2 para as três tensões V_G = -1,0 V; -0,6 V e + 1V.



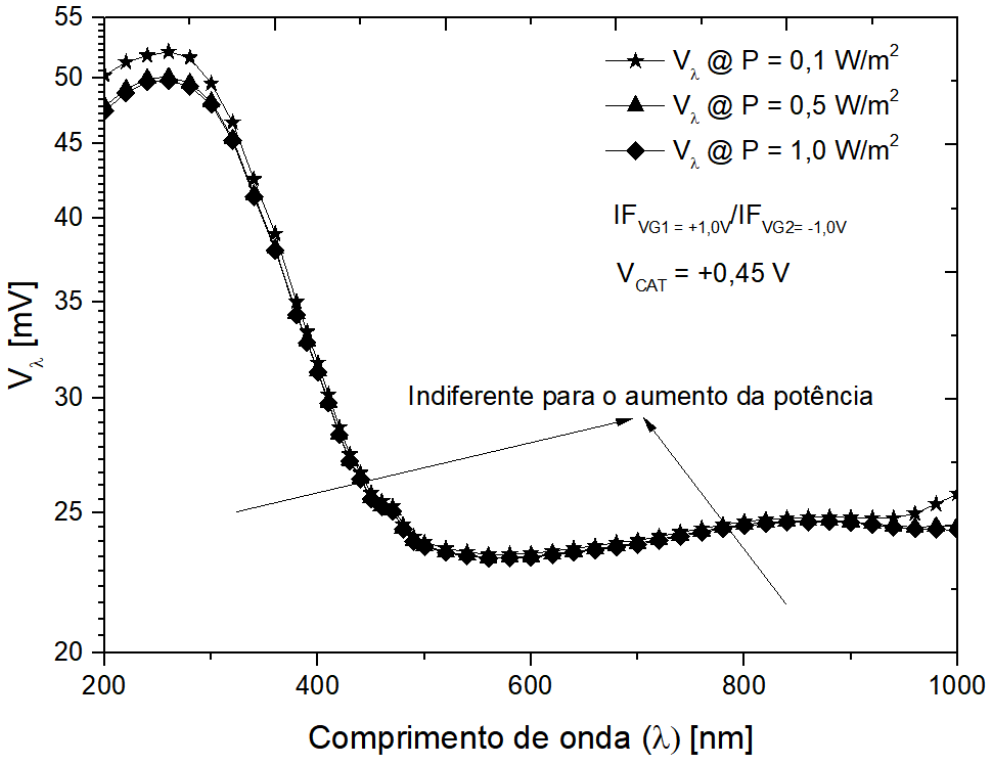
Fonte: Autor

Figura 74 – Comparativo V_λ vs λ com IF_{VG1} = -0,6 V / IF_{VG2} = -1,0 V para as três potências.
.



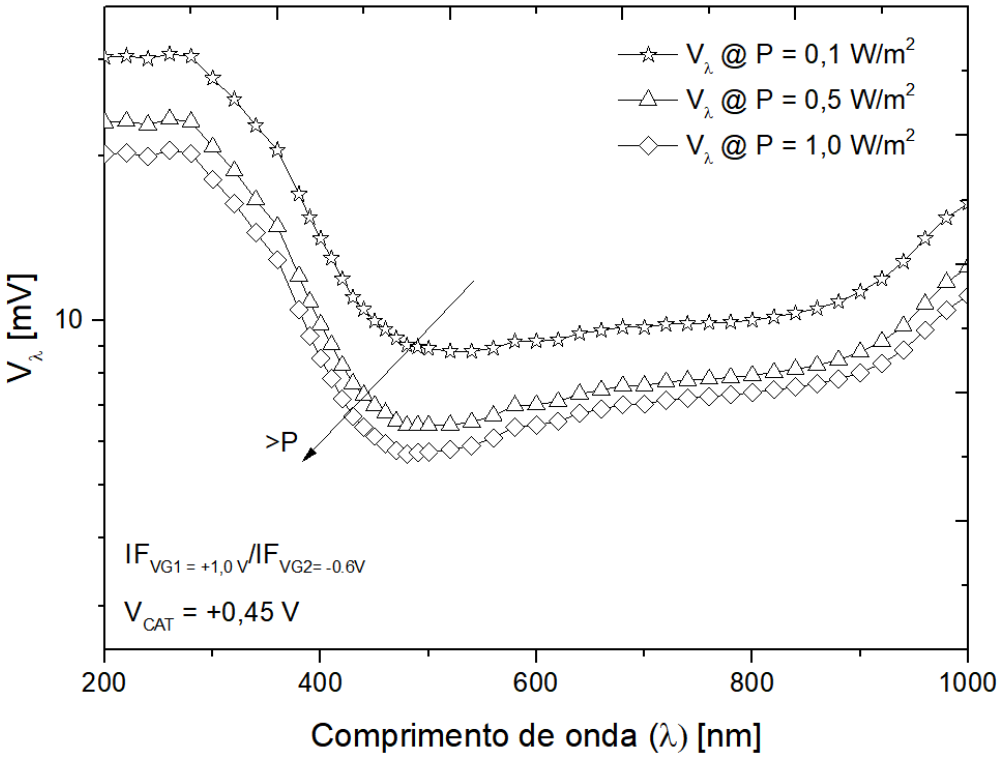
Fonte: Autor

Figura 75 – Comparativo V_λ vs λ com IF_{VG1} = +1,0 V / IF_{VG2} = -1,0 V para as três potências.



Fonte: Autor

Figura 76 – Comparativo V_λ vs λ com IF_{VG1} = +1,0 V / IF_{VG2} = -0,6 V para as três potências.



Fonte: Autor

Para a relação IF_{VG1} = -0,6 V / IF_{VG2} = -1,0 V (Figura 72) mostra que o aumento da potência faz com que V_λ também aumente. Entretanto para a Figura 73, mesmo que a potência aumente, a relação IF_{VG1} = +1,0 V / IF_{VG2} = -1,0 V mostrou-se invariável na tensão de saída V_λ. Já para a relação IF_{VG1} = +1,0 V / IF_{VG2} = -0,6 V, o aumento da potência fez com que a tensão de saída V_λ reduzisse. A Tabela 35 mostra o resumo destas análises.

Tabela 35 – Resumo comparativos das análises entre IF_{VG1} / IF_{VG2} para a tensão de saída V_λ.

V_{G1}	**V_{G2}**	**Faixa λ [nm]**	**Faixa V_λ [mV] @ P = 0,1 W/m²**	**Faixa V_λ [mV] @ P = 0,5 W/m²**	**Faixa V_λ [mV] @ P = 1,0 W/m²**
-0,6V	- 1V	500 a 880	19,82 a 14,82	24,71 a 17,04	27,23 a 17,66
+1V	-1V	500 a 980	50,20 a 23,62	47,78 a 23,50	47,43 a 23,46
+1V	-0,6V	500 a 860	30,37 a 8,79	23,07 a 6,43	20,19 a 5,70

Fonte: Autor

8.1 RESULTADOS PARCIAIS DO CIRCUITO ELETRÔNICO DISCRIMINADOR

A modificação do circuito eletrônico discutido no item 3.6 mostrou-se capaz de discriminar alguns comprimentos de ondas baseadas nas polarizações da tensão de porta proposta e discutidas no capítulo 7. Os resultados mostraram que à medida que V_G aumenta, maior é a corrente de catodo, em virtude da contribuição na taxa de geração de portadores na RDD vertical, entretanto acima de + 0,6 V, a melhoria já não é tão pronunciada, pois o dispositivo alcançou a máxima taxa de geração. Entre as possíveis combinações dos V_G para as três as faixas de potência de 0,1; 0,5 e 1,0 W/m^2, a melhor combinação foi para V_{G1} = +1,0 V pelo V_{G2} = -0,6 V, onde existe uma tensão de V_λ para a faixa dos comprimentos de ondas entre 200 nm a 1.000 nm.

9 CONCLUSÃO

Neste trabalho foi possível constatar um novo caminho de discriminação das cores RGB com auxílio do terminal de porta V_G através de técnicas computacionais de aprendizado de máquina. Foi possível verificar que o terminal de porta tem um papel fundamental para discriminar as cores RGB, contribuindo positivamente na RDD vertical. No capítulo 6 os índices de méritos X_{RGB}, Y_{RGB}, $I'_{CAT}(V_G)$ e $I_{CAT,int}$ demostram essa possibilidade. Para os índices X_{RGB} e Y_{RGB} os estudos mostram que a faixa de V_G se estende de -0,8V à +0,8V. Já para o método derivativo ($I'_{CAT}(V_G)$) a faixa de V_G para discriminar está compreendida entre -0,8V à -0,5V, tendo o ponto máximo de discriminação para V_G = -0,65V. Finalizando os estudos dos índices, o método integrativo ($I_{CAT,int}(V_G)$) informa que a melhor faixa para se trabalhar é 0V à 0,5V, onde o comportamento é linear, apresentando como discriminador os coeficientes angulares (B_X) de cada comprimento de onda RGB. Estabelecidos os critérios de discriminação do fotodiodo PIN lateral com porta, o capítulo 7 trata dos algoritmos gerados pelo aprendizado de máquina, testados e simulados no Matlab 2017. No total foram gerados sete algoritmos computacionais de discriminação baseado nos dados caracterizados de $I_{CAT}(V_G)$ vs V_G discutidos no capítulo 6, sendo: *Linear Discriminant*, *Quadratic Discriminant*, *Linear SVM*, *Quadratic SVM*, *Cubic SVM*, *Cosine KNN* e *Ensemble Subspace Discriminant*. Todos os modelos durante a fase de treinamento mostraram acurácia acima de 90% e ROC/AUC = 1, na primeira rodada. Os modelos que apresentaram as maiores e menores acurácias durante a fase de treinamento e testes respectivamente foram *Quadratic SVM* (99,73% - Treinamento) e o *Linear SVM* (90,32% - Treinamento). Já na fase de teste os modelos que obtiveram maiores e menores médias das acurácias foram *Quadratic SVM* (96,83% - Teste) e o *Cubic SVM* (78,77% - Teste). Os algoritmos gerados e tratados no capítulo 7, mostram um novo formato de análise de dados caracterizados para dispositivos eletrônicos através da interdisciplinaridade das áreas de engenharia elétrica com a engenharia da computação. Por fim, o capítulo 8 mostra a modificação do circuito eletrônico discutido no item 3.6. Essa modificação consiste em retirar os dois fotodiodos PN verticais e substitui-los pelos fotodiodos laterais com porta, sendo que, as polarizações das portas são as mesmas propostas no capítulo 7, que se mostram satisfatórias nos algoritmos do AM. Essas polarizações são quatro no total, -1,0 V, -0,6 V, +0,6 V e +1,0 V, e os resultados demostraram que à medida que o V_G aumenta, a corrente de catodo também aumenta, pois, a polarização da porta contribui de forma positiva na RDD vertical, que em outras palavras, essa polarização melhora a taxa de geração dos portadores dentro da RDD. Entretanto acima de +0,6 V, a corrente de catodo não teve aumentos significativos, concluindo

que o dispositivo, acima dessa polarização está alcançando a sua máxima taxa de geração de portadores. Mediante a essa afirmação, foram propostas três combinações de valores de V_G utilizados nos fotodiodos do circuito eletrônico, no intuito de averiguar qual ou quais são os melhores valores de polarização de discriminação, fazendo com que o circuito, forneça uma tensão de saída V_λ em função dos comprimentos de ondas entre 200 nm a 1.000 nm. Sendo assim, a melhor resposta de detecção foi apresentada para V_{G1} = +1,0 V e V_{G2} = -0,6 V, para as três potências ópticas em análises (0,1; 0,5 e 1,0 W/m^2). As demais combinações demostram resultados satisfatórios para qualquer valor de potência óptica para a faixa de λ entre 200 nm a 500nm, onde existe uma boa variação na tensão de saída (V_λ) do circuito. Fica evidente no transcorrer do trabalho apresentado, que é possível discriminar as cores RGB sem a utilização de filtros ativos, abrindo novos horizontes nas análises de dados caracterizados/simulados de quaisquer dispositivos eletrônicos, o que torna esse trabalho inovador na comunidade cientifica.

PUBLICAÇÕES GERADAS DURANE O PERÍDO DO DOUTORADO

1. NOVO, CARLA; BUHLER, RUDOLF; BAPTISTA, JOAO; GIACOMINI, RENATO ; AFZALIAN, ARYAN ; FLANDRE, DENIS . **Quantum Efficiency Improvement of SOI PIN Lateral Diodes Operating as UV Detectors at High Temperatures**. IEEE Sensors Journal, v. 17, p. 1641-1648, 2017.

2. J.B Júnior; MADEIRA, F.M; GIACOMINI, R. C. **Implementação APS CMOS Global Shutter em Tecnologia 0,35um**. V Simpósio de Pesquisa do Grande ABC, São Bernardo do Campo, 2016.

3. COSTA, P. T. L.; J.B Junior; GIACOMINI, R. C. **Implementação do Microcontrolador em uma FPGA**. Simpósio de Iniciação Científica, Didática e de Ações Sociais de Extensão da FEI, São Bernardo do Campo, 2016.

4. LIMA, T. R.; J.N. JUNIOR; GIACOMINI, R. C. **Simulação APS CMOS Global Shutter em tecnologia 0,13µm**. Simpósio de Iniciação Científica, Didática e de Ações Sociais de Extensão da FEI, São Bernardo do Campo, 2017.

REFERÊNCIAS

A MOSAVI P OZTURK AND K CHAU. Flood Prediction Using Machine Learning Models. **2018**, [s.d.].

ABID, K.; KHOKHAR, A. Z.; RAHMAN, F. Sensors and Actuators A : Physical High responsivity silicon MOS phototransistors. **Sensors & Actuators: A. Physical**, v. 172, n. 2, p. 434–439, 2011.

AFZALIAN, A. Optical Detectors in SOI CMOS Technologies for Blue DVD and Short Distance Optical Communication. p. 195, 2006.

Aplicações Informáticas B. Disponível em: http://arianaesara.blogspot.com.br/2015/. Acesso em: 26 out. 2017.

ASO, A.; KAWANABE, H. DlfogD subtraction. 1982.

ATLAS. User Guide Atlas. [s.d.].

BAKER, J. R. **CMOS. Circuit Design, Layout and Simulation, 2nd Edition**, 1997.

BIGAS, M. et al. Review of CMOS image sensors. **Microelectronics Journal**, v. 37, n. 5, p. 433–451, 2006.

BRIGGS, D. **The Dimensions of Color**. [s.l.] Retrieved, 2007.

BRINK, H.; RICHARDS, J. **Real World Machine Learning**, 2014.

BULTEEL, O. **Silicon-on-Insulator Optoelectronic Components for Micropower Solar Energy Harvesting and Bio-Environmental Instrumentation Jury**. Louvain-la-Neuve: [s.n.].

CAMPOS, F. D. **Sistema de imagem CMOS com alta responsividade**. Campinas: Unicamp, 2008.

CHAUHAN, T. et al. Supervised and Unsupervised Machine Learning based Review on Diabetes Care. **2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021**, p. 581–585, 2021.

CHMIELEWSKI, A. M. M. **ANÁLISE E PROJETO DE UM SENSOR DE IMAGEM EM 0.35um CMOS PARA COMPRESSÃO DE DADOS NO PLANO FOCAL DE CÂMERAS DIGITAIS**. Mestrado e ed. Rio de Janeiro: COPPE/UFRJ, 2009.

CIE 1931 Color Space. Disponível em: https://en.wikipedia.org/wiki/CIE_1931_color_space. Acesso em: 27 out. 2017.

CMYK e RGB. Disponível em: https://pura-barbaridade.blogspot.com.br/2010/03/cemik.html. Acesso em: 27 out. 2017.

COLINGE, J. P.; COLINGE, C. **Physics of semiconductor devices**. Massachusetts: Kluwer Academic, 2002.

Cor. Disponível em: http://corisectelmo.blogspot.com.br/2010/11/. Acesso em: 26 out. 2017.

ELANI, U. A. I. The effective carrier lifetime measurement in silicon : The conductivity modulation method. **Journal of King Saud University - Science**, v. 22, n. 1, p. 9–13, 2010.

EVERSON, R. M. AND FIELDSEND, J. E. Multi-class roc analysis froma multi-objective optimisation perspective. Pattern Recognition Letters. 2006.

FARIAN, Ł.; LEÑERO-BARDALLO, J. A.; HÄFLIGER, P. A Bio-Inspired AER Temporal Tri-Color Differentiator Pixel Array. **IEEE Transactions on Biomedical Circuits and Systems**, v. 9, n. 5, p. 686–698, 2015.

Filtro Bayer. Disponível em: http://wikipedia.qwika.com/en2pt/Bayer_filter. Acesso em: 27 out. 2017.

FINDLATER, K. M. et al. A CMOS image sensor employing a double junction photodiode. **IEEE Workshop on Charge-Coupled Devices & Advanced Image Sensors**, p. 60–63, 2001.

F.M, M. S. R. G. S. M. DE; A. P. G. D. G. R. C. SOI Global-Shutter Sample-and-Hold Circuit for Image Sensors compared to Bulk CMOS and nMOS. n. V, p. 2017, 2011.

FOVEON; MERRILL, R. B. **Color Separation in an Active Pixel Cell Imaging Array using a triple well structure**. United State: Patent n. 5 965 875, 1999.

GALAL, M. et al. Characterization of RGB LEDs as Emitter and Photodetector for LED-to-LED Communication. **2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2020**, p. 5–10, 2020.

GAMAL, A. EL. Abbas El Gamal and Helmy Eltoukhy. n. June, 2005.

GIMENES GRADOS, H. R. DESENVOLVIMENTO DE UM PROCESSO CMOS (2 μ m): FABRICAÇÃO DO CHIP TESTE CMOS , CÉLULAS APS Banca examinadora : 2003.

HAGIWARA, Y. Simulation and Device Characterization of the P+PN+P Junction Type Pinned Photodiode and Schottky Barrier Photodiode. **4th Electron Devices Technology and Manufacturing Conference, EDTM 2020 - Proceedings**, p. 8–11, 2020.

HAND, D. J. AND TILL, R. J. A simple generalization of the area under the roc curve to multiple class classication problem. Machine Learning. 2001.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **Machine Learning Book**. [s.l: s.n.]. v. 27

HIRSCH, R. **Exploring Dimensions Colour Photography**. [s.l.] Laurence King, 2004.

I O FREEMAN A J HAIGLER S E SCHMEELK L R ELLRODT AND T L. Filds 2018 What are they Researching? Examining Industry-Based Doctroal Dissertation Research through the

Lens of Machine Learning IEEE International Conference on Machine Learning and Application. [s.d.].

JACOBUS, W. S. **Materiais Elétricos**. [s.l: s.n.].

KAWADA, S. et al. Checkered white-RGB color LOFIC CMOS image sensor. **Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC**, p. 347–348, 2010.

KIM, H. S. et al. Characteristics of a fabricated PIN photodiode for a matching with a CsI(Tl) scintillator. **IEEE Transactions on Nuclear Science**, v. 57, n. 3 PART 2, p. 1382–1385, 2010.

LEÑERO-BARDALLO, J. A.; BRYN, D. H.; HÄFLIGERR, P. Bio-inspired asynchronous pixel event tricolor vision sensor. **IEEE Transactions on Biomedical Circuits and Systems**, v. 8, n. 3, p. 345–357, 2014.

LIMA, K. G. D. **Estruturas APS Resistentes à Radiação para Aplicações Espaciais**. Mestrado e ed. Rio de Janeiro: COPPE/UFRJ, 2006.

LONGONI, A. et al. The Transverse Field Detector: a novel color sensitive CMOS Device. **Electron Device Letters**, v. 29, p. 12, 2008.

LU, M.; LI, F. Survey on lie group machine learning. **Big Data Mining and Analytics**, v. 3, n. 4, p. 235–258, 2020.

MADEIRA, F. M. ESTUDO COMPARATIVO ENTRE TECNOLOGIAS CMOS, NMOS E SOI EM UM CIRCUITO APS. 2012.

MARTINO, J. A.; PAVANELLO, M. A.; VERDONCK, P. B. **Caracterização elétrica de tecnologia e dispositivos MOS**. São Paulo: Thonsom, 2003.

MATILAB. **Matlab User Guide**. [s.l: s.n.].

MENON, P. S. et al. High performance of a SOI-based lateral PIN photodiode using SiGe/Si multilayer quantum well. **2012 10th IEEE International Conference on Semiconductor Electronics, ICSE 2012 - Proceedings**, p. 403–406, 2012.

MOGHAVVEMI, M. et al. Design of low cost flexible RGB color sensor. **2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012**, p. 1158–1162, 2012.

MUTTAQIN, J.; WIJAYA, S. K.; PRAJITNO, P. The design of two dimensional gamma ray tomography system hardware based-on PIN photodiode detector. **2018 International Conference on Signals and Systems, ICSigSys 2018 - Proceedings**, p. 236–242, 2018.

NAKAMURA, J. **Image Sensor and Signal Processing for Digital still Cameras**. [s.l.] CRC, 2005.

NEAMEN A., D. **Semiconductor Physics and Devices: Basic Principles**. McGraw-Hil ed. [s.l: s.n.].

Novel Sensors Help Smart Cameras Serve Niches. Disponível em: http://www.vision-systems.com/articles/print/volume-17/issue-5/features/novel-sensors-help-smart-cameras-serve-niches.html. Acesso em: 27 out. 2017.

NOVO, C. **PROJETO E CARACTERIZAÇÃO DE DIODOS PIN LATERAIS-TOPOLOGIAS, CARACTERÍSTICAS E PROPOSTA DE UMA NOVA CONFIGURAÇÃO**. [s.l: s.n.].

NOVO, C. D. **Estudo comparativo de tecnologias CMOS e SOI-CMOS aplicadas a fotodiodos laterais PIN**. [s.l.] Centro Universitário da FEI, 2013.

RAVAZI, B. **Fundamentals of Microeletronics**. [s.l.] LTC - GRUPO GEN, 2010.

ROSS, M. et al. Using support vector machines to classify student attentiveness for the development of personalized learning systems. **Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013**, v. 1, p. 325–328, 2013.

SAKAI, S. et al. A pixel-shared CMOS image sensor using lateral overflow gate. **ESSCIRC 2009 - Proceedings of the 35th European Solid-State Circuits Conference**, p. 240–243, 2009.

SEAUENCE, A. L. et al. Three-Color Sensor Based on. v. 42, n. 10, p. 1763–1768, 1995.

SHI, L. ET AL. Electrical and Optical Performance Investigation of Si- Based Ultrashallow Juction P+n-VUV/EUV Photodiodes. p. 1268–1271, 2012.

Smart CMOS Image Sensors. [s.l: s.n.].

STREEMAN, B. G.; BANERJEE, S. **Solid state electronic devices**. New Jersey: Prentice Hall, 2000.

SZE, M. S. **Physics os semiconductor devices**. New York: John Wiley & Sons, 1981.

T. G. DIETTRICH. Machine Learning Research. **vol. 18**, p. pp.97-136, 1997.

VILARINO, F., KUNCHEVA, L. I., AND RADEVA, P. Roc curves and video analysis optimization in intestinal capsule endoscopy. Pattern Recognition Letters. p. pages 875 881, 2006.

VONRUEDEN, L. et al. Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. **IEEE Transactions on Knowledge and Data Engineering**, p. 1–20, 2021.

WONG, H. P. et al. CMOS active pixel image sensors fabricated using a 1.8-V, 0.25-µm CMOS technology. **IEEE Transactions on Electron Devices**, v. 45, n. 4, p. 889–894, 1998.

WYSZECKI, G. **COLOR SCIENCE**. 2º ed. Canada: John Wiley & Sons', 1982.

Y. QIAN, W. ZHOU, J. YAN, W. LI, L. H. Comparing Machine Learning Classifiers for Object-Based Land Cover Classification Using Very High Resolution Imagery. p. pp.154-168, 2015.

YANG, J. et al. A 320x240-pixel Sub-centimeter Resolution Time-of-Flight Image Sensor Design with Structure and Doping Optimized Pinned Photodiode Pixels. **2019 IEEE International Conference on Integrated Circuits, Technologies and Applications, ICTA 2019 - Proceedings**, p. 174–175, 2019.

YU, J. et al. An incremental learning approach for restricted boltzmann machines. **ICCAIS 2015 - 4th International Conference on Control, Automation and Information Sciences**, p. 113–117, 2015.

ZIMAN, M. J. **Principles of theory of solids**. 2nd Editio ed. [s.l.] Cambridge University Press, 1972.

ZIMMERMANN, H. **Integrated Silicon Opto-electronics**. Berlin: Springer, 2000.

APÊNDICE A - PROGRAMA PARA TESTAR OS ALGORITMOS AM

```
%************Script to predict Photodiode****************
load Esemble_Discriminant
data_test=readtable("C:\Users\batis\Desktop\Simulacao_PIN\ML_PIN_BULK_NOVO\
Dados_PIN_BULK_TESTE_FUTURE_4G.xlsx");
data_train=readtable("C:\Users\batis\Desktop\Simulacao_PIN\ML_PIN_BULK_NOVO
\Dados_PIN_BULK_TESTE_ATUAL_4G.xlsx");

%-------------------------------------------------------------
%Script to predict RGB color of PIN Gated Photodiode
%-------------------------------------------------------------
%load Diode_Teste % Load the computational model trainedCLC
%load data_test % Load new data to prediction
actual_data=data_train.CLASSE; % Load actual data of training
%Execute the prediction and compare with actual data
variable_predict=Esemble_Discriminant.predictFcn(data_test);
compare_table=table(variable_predict,actual_data)
h=heatmap(compare_table,'variable_predict','actual_data');
Accuracy_Percent=(sum(compare_table.variable_predict==compare_table.actual_
data)/height(compare_table))*100
h.Title=(['Predict Data vs Acutal Data --------- Accuracy :
',num2str(Accuracy_Percent),' (%)']);
h.XLabel = 'Predict Data';
h.YLabel = 'Actua Data';

confMat=[11,1,0;0,12,0;0,1,11];

for i =1:size(confMat,1)
    precision(i)=confMat(i,i)/sum(confMat(i,:));
end
precision(isnan(precision))=[];
Precision=sum(precision)/size(confMat,1)

for i =1:size(confMat,1)
    recall(i)=confMat(i,i)/sum(confMat(:,i));
end

Recall=sum(recall)/size(confMat,1)

F_score=2*Recall*Precision/(Precision+Recall)
```

APÊNDICE B- ALGORITMO *LINEAR DISCRIMINANT*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:17:25


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationDiscriminant = fitcdiscr(...
```

```
    predictors, ...
    response, ...
    'DiscrimType', 'linear', ...
    'Gamma', 0, ...
    'FillCoeffs', 'off', ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
discriminantPredictFcn = @(x) predict(classificationDiscriminant, x);
trainedClassifier.predictFcn = @(x)
discriminantPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationDiscriminant = classificationDiscriminant;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationDiscriminant,
'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE C- ALGORITMO *QUADRATIC DISCRIMINANT*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:18:50


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationDiscriminant = fitcdiscr(...
    predictors, ...
    response, ...
```

```
    'DiscrimType', 'quadratic', ...
    'FillCoeffs', 'off', ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
discriminantPredictFcn = @(x) predict(classificationDiscriminant, x);
trainedClassifier.predictFcn = @(x)
discriminantPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationDiscriminant = classificationDiscriminant;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationDiscriminant,
'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE D - ALGORITMO *LINEAR SVM*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:19:51


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'linear', ...
    'PolynomialOrder', [], ...
```

```
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    predictors, ...
    response, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x)
svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationSVM, 'KFold',
5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE E - PROGRAMA *QUADRATIC SVM*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:20:46


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'polynomial', ...
    'PolynomialOrder', 2, ...
```

```
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    predictors, ...
    response, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x)
svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationSVM, 'KFold',
5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE F - ALGORITMO *CUBIC SVM*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:22:16


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
template = templateSVM(...
    'KernelFunction', 'polynomial', ...
    'PolynomialOrder', 3, ...
```

```
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true);
classificationSVM = fitcecoc(...
    predictors, ...
    response, ...
    'Learners', template, ...
    'Coding', 'onevsone', ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x)
svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationSVM, 'KFold',
5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE G - ALGORITMO *COSINE KNN*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:23:24


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
classificationKNN = fitcknn(...
    predictors, ...
    response, ...
```

```
    'Distance', 'Cosine', ...
    'Exponent', [], ...
    'NumNeighbors', 10, ...
    'DistanceWeight', 'Equal', ...
    'Standardize', true, ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
knnPredictFcn = @(x) predict(classificationKNN, x);
trainedClassifier.predictFcn = @(x)
knnPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationKNN = classificationKNN;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationKNN, 'KFold',
5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```

APÊNDICE H - ALGORITMO *ENSEMBLE SUBSPACE DISCRIMINANT*

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] = trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code recreates the
% classification model trained in Classification Learner app. Use the
% generated code to automate training the same model with new data, or to
% learn how to programmatically train models.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedClassifier: a struct containing the trained classifier. The
%       struct contains various fields with information about the trained
%       classifier.
%
%      trainedClassifier.predictFcn: a function to make predictions on new
%       data.
%
%      validationAccuracy: a double containing the accuracy in percent. In
%       the app, the History list displays this overall accuracy score for
%       each model.
%
% Use the code to train the model with new data. To retrain your
% classifier, call the function from the command line with your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the original data set
% T, enter:
%   [trainedClassifier, validationAccuracy] = trainClassifier(T)
%
% To make predictions with the returned 'trainedClassifier' on new data T2,
% use
%   yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 28-Mar-2021 14:24:32


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Train a classifier
% This code specifies all the classifier options and trains the classifier.
subspaceDimension = max(1, min(2, width(predictors) - 1));
classificationEnsemble = fitcensemble(...
    predictors, ...
    response, ...
```

```
    'Method', 'Subspace', ...
    'NumLearningCycles', 30, ...
    'Learners', 'discriminant', ...
    'NPredToSample', subspaceDimension, ...
    'ClassNames', categorical({'AZUL'; 'VERDE'; 'VERMELHO'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
ensemblePredictFcn = @(x) predict(classificationEnsemble, x);
trainedClassifier.predictFcn = @(x)
ensemblePredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VG_N_1', 'VG_N_06', 'VG_P_06',
'VG_P_1'};
trainedClassifier.ClassificationEnsemble = classificationEnsemble;
trainedClassifier.About = 'This struct is a trained model exported from
Classification Learner R2017a.';
trainedClassifier.HowToPredict = sprintf('To make predictions on a new
table, T, use: \n  yfit = c.predictFcn(T) \nreplacing ''c'' with the name
of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table,
T, must contain the variables returned by: \n  c.RequiredVariables
\nVariable formats (e.g. matrix/vector, datatype) must match the original
training data. \nAdditional variables are ignored. \n \nFor more
information, see <a href="matlab:helpview(fullfile(docroot, ''stats'',
''stats.map''), ''appclassification_exportmodeltoworkspace'')">How to
predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'VG_N_1', 'VG_N_06', 'VG_P_06', 'VG_P_1'};
predictors = inputTable(:, predictorNames);
response = inputTable.CLASSE;
isCategoricalPredictor = [false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedClassifier.ClassificationEnsemble,
'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] = kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun',
'ClassifError');
```