Dissertação de Mestrado

Classificação de Rasuras em Textos Manuscritos

Marcos Antônio Rocha Tenório
marcos@ic.ufal.br

Orientadores:

Prof. Dr. Evandro de Barros Costa
Prof. Dr. Tiago Figueiredo Vieira

Maceió, Fevereiro de 2018

Marcos Antônio Rocha Tenório

Classificação de Rasuras em Textos Manuscritos

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Universidade Federal de Alagoas – UFAL

Instituto de Computação

Pós-Graduação em Modelagem Computacional do Conhecimento.

Orientador: Prof. Dr. Evandro de Barros Costa

Coorientador: Prof. Dr. Tiago Figueiredo Vieira

Maceió

Fevereiro 2018

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Bibliotecária Responsável: Janis Christine Angelina Cavalcante

T289c Tenório, Marcos Antônio Rocha.
Classificação de rasuras em textos manuscritos / Marcos Antônio Rocha Tenório. – 2018.
82f. : il., grafs., tabs.

Orientador: Evandro de Barros Costa.
Coorientador: Tiago Figueiredo Vieira.
Dissertação (Mestrado em Modelagem Computacional de Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Programa de Pós-Graduação em Modelagem Computacional de Conhecimento. Maceió, 2018.

Bibliografia: f. 63-64.
Apêndices: f. 66-73.
Anexos: f. 75-82

1. Modelagem computacional. 2. Texto manuscrito – Identificação de rasuras. 3. Linguística. I. Título.

CDU: 004.9:81

UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 12, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364/1825

Membros da Comissão Julgadora da Dissertação de Mestrado de Marcos Antônio Rocha Tenório, intitulada: "Classificação de Rasuras em Textos Manuscritos", apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas, em 8 de fevereiro de 2018, às 15h00min, na sala de reuniões do Instituto de Computação da Ufal.

COMISSÃO JULGADORA

Prof. Dr. Evandro de Barros Costa
Ufal – Instituto de Computação
Orientador

Prof. Dr. Tiago Figueiredo Vieira
Ufal – Instituto de Computação
Coorientador

Prof. Dr. Aydano Pamponet Machado
Ufal – Instituto de Computação
Examinador

Prof. Dr. Eduardo Calil de Oliveira
Ufal – Centro de Educação
Examinador

Prof. Dr. Joilson Batista de Almeida Rêgo
UFRN – Escola de Ciências e Tecnologia
Examinador

Maceió, fevereiro de 2018.

Aos meus filhos Marcus, Matheus, Maria Manuela e Maria Eduarda
e aos meus irmãos Ignez, Tânia, Lúcia, Jásson, Alice e Selma,
pelo tempo que deixamos de estar juntos e
por todo o tempo que estivemos juntos...
Aos meus pais, José e Tércia,
à eles todos os créditos...

Agradecimentos

Agradeço aos meus pais e aos meus irmãos por sempre orientarem o meu caminho e me terem feito perceber o quanto importante era a estrada do conhecimento. Pelo constante apoio e respeito às minhas decisões e escolhas (“Nunca lhe diremos o que fazer, mas sempre lhe apoiaremos naquilo que você decidir! ”). Agradeço aos meus filhos pela oportunidade de ser pai, de saber que o pouco que eu sei poderá ser multiplicado e replicado para outros através deles. Por me apoiarem e valorizarem sempre. Agradeço aos outros membros da minha família, sobrinhos e agregados pelo carinho e paciência e respeito as minhas ausências. Agradeço à minha nora predileta pela revisão do texto e tradução de outros. Agradeço à minha companheira pela Moraes, digo moral, dada durante todo esse processo. Agradeço ao meu orientador professor doutor Evandro de Barros Costa e ao professor doutor Tiago Figueiredo Vieira pelos conselhos dados e por me convencer a levar essa jornada até o fim. Agradeço também a todos os meus professores, principalmente os do ensino fundamental, que foram os responsáveis por colocar os primeiros tijolos na construção do meu conhecimento e me ajudaram a ser o que sou. Agradeço ao LAME - Laboratório do Manuscrito Escolar, pelo fornecimento das imagens do seu acervo, que nos ajudaram a exemplificar e categorizar as rasuras. Por fim, agradeço também a todos que me ajudaram a concluir esse trabalho, fosse com carinho, com atenção, com apoio moral e logístico, com pão e com circo!

"The only way of finding the limits of the possible is by going beyond them into the impossible."

(Arthur C. Clarke)

Resumo

Este trabalho se insere na linha de pesquisa Modelos Computacionais em Educação do Programa de Pós-graduação Interdisciplinar em Modelagem Computacional de Conhecimento, realizando um estudo sobre classificacão de rasuras em textos manuscritos. O estudo dessas rasuras é feito para a determinação das dificuldades no processo de aprendizagem, pois leva a entender o conflito do estudante em processo de aquisição da escrita, na identificação das fronteiras da palavra ortográfica. Neste contexto, o objetivo deste trabalho foi a construção de um modelo para identificação e detecção das rasuras em textos, através da análise das rasuras geradas em palavras de enunciados produzidos pelos referidos estudantes. Para representar matematicamente as imagens, devido à diferença de dimensões das imagens adquiridas, fez-se necessário a utilização de um parâmetro que caracterizasse a imagem de uma forma única, independente da mesma ser uma letra ou palavra, ou das dimensões dos caracteres utilizados na escrita. Na solução deste problema, propõe-se a identificação da direção dos traços que compõem a escrita, pois a mesma independe das dimensões e caracteriza bem principalmente as rasuras, normalmente composta por muitos traços paralelos em uma única direção. As imagens foram classificadas por processos de aprendizagem de máquina, das técnicas: Máquina de Vetores de Suporte (SVM), Árvores de Decisão (*boosted and bagged trees*), Vizinhos mais Próximos (KNN), *Naive Bayes*, análise de discriminante (QDA e outros) e Regressão Logística, que analisam os dados e reconhecem padrões, usados para classificação e análise de regressão. Os resultados obtidos mostram que os descritores de imagem quando utilizados isoladamente obtém valores limites de acurácia na ordem de 90%, porém quando combinados com outros (Histograma de Orientação de Borda + Centróide Hierárquico + Momentos de Zernike), excedem 98% de acurácia. Assim, uma das conclusões obtidas nessa pesquisa é a da apropriação de técnicas de aprendizagem de máquina para o reconhecimento de rasuras.

Palavras-chave: classificação. identificação de rasuras. linguística.

Abstract

This work is included in the research line "Computational Models in Education" of the Interdisciplinary Postgraduate Program in Computational Modeling of Knowledge, conducting a study on erasures in texts produced by students of primary education. This study is performed aiming to detect difficulties in the learning process, because it leads to understand the conflict of the student in the process of acquisition of writing, in the identification of the orthographic word boundaries. In this context, the objective of this paper is the construction of a model for identification and detection of erasures in texts, through the analysis of erasures generated in words of statements produced by students. In order to represent the images mathematically, due to the difference in the dimensions of the images acquired, it was necessary to use a parameter that characterized the image in a unique way, regardless of whether it is a letter or word, or the dimensions of the characters used in writing. In the solution of this problem, it is proposed to identify the direction of the traces that compose the writing, since it is independent of the dimensions and characterizes mainly the erasures, usually composed by many parallel traces in a single direction. The images were classified by machine learning processes, using techniques such as: Support Vector Machines (SVM), Boosted and bagged trees, Nearest Neighbors (KNN), Naive Bayes, discriminant analysis (QDA and others) and Logistic Regression, which analyze the data and recognize used for classification and regression analysis. The obtained results show that the image descriptors when used alone obtains limits around 90% of accuracy, but when combined with others (Edge Orientation Histogram + Hierarchical Centroid + Zernike Moments), they exceed 98% of accuracy. Thus, one of the conclusions obtained in this research is the appropriation of machine learning techniques for the recognition of erasures.

Keywords: Classification. Identification of Erasures. Linguistics.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

UFAL	Universidade Federal de Alagoas
IFAL	Instituto Federal de Alagoas
NIST	*National Institute of Standards and Technology*
ROC	*Receiver Operating Characteristics* - Características operacionais do receptor
LoG	*Laplacian of Gaussian* - Laplaciano da Gaussiana
PCA	*Principal Component Analysis* - Análise da Componente Principal
LDA	L*inear Discriminant Analysis* - Análise da Componente Principal
SVM	*Support Vector Machine* - Máquina de Vetor de Suporte
KNN	*K-Nearest Neighbors* - k-vizinhos mais próximos
QDA	*Quadratic Discriminant Analysis* - Análise de Discriminante Quadrático
S./C.	Segmentos/Características
SVM L.	SVM Linear
S.RBF	SVM de Função de Base Radial
P.Gaus.	Processo Gaussiano
Arv.Dec.	Árvore de Decisão
Flo.Ale.	Floresta Aleatória
R. N.	Rede Neural
Adab.	Adaboost
N.Bayes	*Naive Bayes*
LAME	Laboratório do Manuscrito Escolar
FAPEAL	Fundação de Amparo à Pesquisa do Estado de Alagoas
NExOS	Núcleo de Excelência em Otimização de Sistemas Complexos

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Sumário

1 Introdução

O estudo da linguagem escrita dentro do processo de aprendizagem é importante para determinar as dificuldades enfrentadas pelo educando no processo de transformação da linguagem falada em linguagem escrita. Durante este processo, o educando comete falhas e tenta corrigi-as de imediato, seja através de um processo de riscar a letra/palavra errada, apagar e reescrever ou mesmo sobrepor o escrito errado pelo correto, dentre outras alternativas, marcações e anotações. Essas tentativas de correções, chamadas de rasura – podem ser entendidas como lugares de conflito, que permitem visualizar o sujeito escrevente num movimento de negociação e de estranheza de sua própria palavra ((CALIL, 2008); (FELIPETO, 2008); (CAPRISTANO, 2007)), ou ainda, numa tentativa de 'ajuste' que dá saliência ao enlaçamento entre o falado/oral e o escrito/letrado e à própria "heterogeneidade constitutiva da escrita" ((CôRREA, 2004); (CAPRISTANO; CHACON, 2014)). Os primeiros estudos sobre rasuras datam do final dos anos 60 e surgiram na perspectiva da Crítica Genética, e buscavam compreender o processo da elaboração do texto, a partir do prototexto, ou seja, tudo que se escreve antes ou em vista do texto a ser publicado ((WILLEMART, 1993)), ou seja, rascunhos, manuscritos e suas rasuras. Além dessa abordagem, outros estudos são feitos por linguistas, onde as rasuras, chamadas de reescrita, reelaboração ou refacção, são analisadas sob uma perspectiva de aquisição da escrita. Essa dissertação surgiu dentro do grupo de pesquisa NExOS - Núcleo de Excelência em Otimização de Sistemas Complexos, em um contexto de projeto contemplado em um edital Pronex, financiado pela FAPEAL - Fundação de Amparo à Pesquisa do Estado de Alagoas.

1.1 Problemática

Para o estudo dessas rasuras ser efetuado, independente da abordagem e do tipo de análise a ser feito, amostras de textos manuscritos devem ser coletadas, letras e palavras rasuradas observadas e categorizadas. Este processo é manual e não padronizado em relação à identificação das amostras obtidas. Pessoas distintas podem categorizar uma rasura de maneiras diversas, grafias diferentes podem levar à uma interpretação errada do tipo de rasura analisada ou mesmo classificação de uma rasura como não-rasura (texto normal). Assim, ressalta-se um desafio: Como identificar e categorizar essas rasuras?

1.2 Objetivo

Considerando a problemática exposta, esse trabalho tem como objetivo, o desenvolvimento de um modelo computacional, que possa identificar, discriminar e categorizar essas amostras através de mecanismos de aprendizagem de máquina, tornando a tarefa de reconhecimento das amostras rápida e padronizada. A velocidade dessa operação permitirá não só a classificação das rasuras, mas também, através do estudo de sua forma, quantidade, distribuição ao longo do texto e outras estatísticas, tentar descobrir o que gera esses "momentos de conflito" e conseqüentemente, a própria rasura.

1.3 Relevância

Espera-se, a partir do desenvolvimento deste trabalho, a obtenção dos seguintes ganhos:

- obtenção de um modelo que possa identificar e classificar os tipos de rasuras presentes em um manuscrito de uma forma rápida e eficiente.
- A utilização desse modelo para levantamento de características e descritores dentro do texto manuscrito, que possam levar a conclusão do que ocasiona a rasura.
- O estudo de casos conhecidos de distúrbios que levam à produção de rasuras e a possível inter-relação entre os tipos de rasuras e os mesmos. Considerando a quantidade de dados que podem ser levantados por um processo totalmente automatizado, quaisquer outros processos de análise estatística dos textos sobre estudo podem ser melhorados e aperfeiçoados.

1.4 Estrutura do Trabalho

Este trabalho foi dividido em capítulos da seguinte maneira:

- Capítulo 2 – A fundamentação teórica em que se baseia este trabalho, e trabalhos relacionados. São apresentados conceitos sobre rasuras, os descritores de imagem testados e o utilizado além de noções sobre aprendizado de máquina e o software onde foram realizados os testes, além dos trabalhos relacionados.
- Capítulo 3 – Proposta do Classificador - Onde serão vistos os classificadores utilizados e os ajustes feitos para utilização dos mesmo.
- Capítulo 4 – Resultados e Discussões - Os resultados obtidos para os algoritmos dos classificadores utilizados e uma discussão acerca dos mesmos.

- Capítulo 5 - Conclusões - Discussão dos aspectos relevantes do trabalho e sugestões para trabalhos futuros.

2 Fundamentação Teórica e Trabalhos Relacionados

Este capítulo tem como objetivo a apresentação dos conceitos teóricos utilizados neste trabalho. Serão mostrados os conceitos da rasura como objeto de estudo linguístico e como imagem a ser analisada, descritores de imagem, sistemas de aprendizado e máquina (classificadores) e as ferramentas utilizadas. Inicialmente serão mostrados conceitos lingüísticos relacionados à rasura, descritores de imagem como elementos de caracterização da imagem como única, os sistemas de aprendizagem de máquina e sua terminologia, os classificadores e a escolha do mesmo em função das características apresentadas pelas imagens e o sciki-learn (Python) como ferramenta de implementação do modelo e para simulação de testes.

2.1 Rasuras

Rasuras são marcações gráficas feitas pelo escrevente para a substituição de textos manuscritos em decorrência de diversos fatores: o erro na escrita, uma reformulação do pensamento, uma melhoria na ideia a ser transmitida, etc. Normalmente marcam um ponto de conflito entre aquilo que o escrevente pensa e aquilo que quer transmitir. Do ponto de vista lingüístico a geração dessas rasuras pode ocorrer através de quatro operações: a supressão, a substituição, a adição e o deslocamento. Na supressão o termo escolhido é riscado, rasurado e não é substituído. Caracteriza uma das mais radicais operações pois o autor não modifica o texto, abandona-o por completo. Segundo (FELIPETO, 2008), "diante de uma dificuldade, o sujeito decide suprimir o fragmento e abandoná-lo". Na substituição, o termo rasurado é trocado por outro ou por ele mesmo. Na adição, algum elemento, seja uma letra ou uma palavra completa, é inserida no texto depois de uma releitura. O deslocamento acontece quando um elemento escrito, seja ele uma letra, sílaba ou palavra, é transferido de uma posição para outra. Segundo (CALIL, 2008), essas operações podem ainda se manifestar de várias formas:

- Rasura "riscada": caso em que se anula o que se escreveu, de modo geralmente visível, permitindo ao leitor recuperar o texto rasurado. Esta forma também poderia aparecer como "rasura apagada", quando o autor apaga com uma borracha, mas deixa os vestígios que permitem ler o que havia escrito.
- Rasura "borrão": casos em que se anula o que foi escrito, mas que não é permitido ler o que foi rasurado, cobrindo todo o escrito ou parte dele com uma mancha de

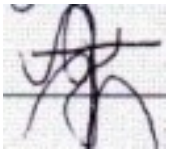


Figura 1 – Rasura “riscada” - Acervo LAME

tinta opaca ou ainda apagando completamente os traços do escrito.

Figura 2 – Rasura “borrão” - Acervo LAME

- Rasura “branca” ou “imaterial”: somente se tem acesso a ela pela comparação de versões sucessivas de um manuscrito, pois o autor a produz enquanto copia a versão anterior. (CALIL, 2008)

  Ainda para (CALIL, 2008), “do ponto de vista do espaço da folha de papel, as rasuras podem aparecer como:

- Rasura “linear”: aquelas apresentadas em uma mesma linearidade, em que um elemento (letra, palavra, frase, parágrafo) pode ser riscado e imediatamente reescrito na continuidade da linha.

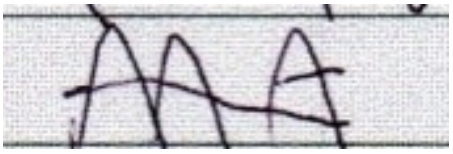


Figura 3 – Rasura “linear” - Acervo LAME

- Rasura “sobreescrita”: em geral incide sobre uma letra, uma parte da palavra ou sobre palavras curtas, em que se escreve sobre aquilo que já estava gravado.

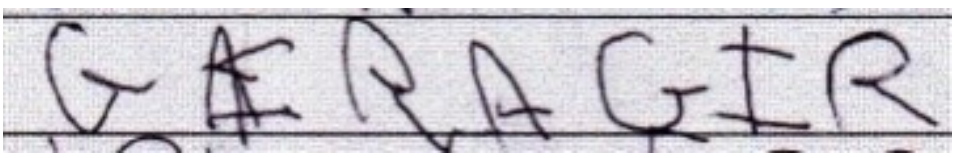


Figura 4 – Rasura “sobreescrita” - Acervo LAME

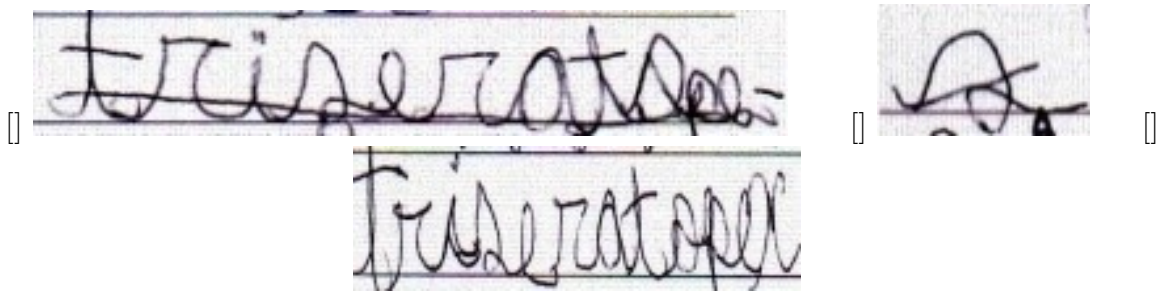


Figura 5 – Rasuras interlineares - Acervo LAME

- Rasura "interlinear": marcas feitas entre o espaço de duas linhas.
- Rasura "marginal": escritos que aparecem à margem do texto, podendo referir-se um comentário sobre o que foi escrito ou propor a entrada de um novo elemento ao texto; geralmente elas são indicadas por setas, chaves, asteriscos, números e são produzidas após a escrita do texto. (CALIL, 2008)

O algoritmo visa reconhecer a maioria das rasuras e classificá-las. A partir desta classificação, obter dados que indiquem informações relevantes sobre o processo da aquisição da linguagem escrita.
Todas as imagens de rasuras mostradas aqui pertencem ao acervo do LAME - Laboratório do Manuscrito Escolar.

2.2 Extração de Características da Imagem

O objectivo da extração de características é a medição dos atributos de padrões que são mais pertinentes para uma dada tarefa de classificação. A tarefa do especialista humano é selecionar ou inventar recursos que permitem o reconhecimento eficaz e eficiente dos padrões. Muitos recursos têm sido descobertos e utilizados para o reconhecimento de padrões.

2.2.1 Extração de Características

As características podem ser classificadas em características geométricas, características estruturais e características de métodos de transformações espaciais. As características geométricas incluem momentos, histogramas e características de direção. As características estruturais incluem o registro, características de elementos de linha, os descritores de Fourier, características topológicas, e assim por diante. Os métodos de transformação incluem análise de componente principal (PCA), análise discriminante linear (LDA), PCA do kernel, transformada de Fourier e transformada Wavelet, dentre outros. Veremos aqui uma definição rápida de algumas que foram utilizadas em nosso trabalho.

2.2.2 Momentos

Momentos e funções de momentos têm sido empregados como padrão de características em numerosas aplicações para reconhecer padrões em imagens bidimensionais. Estas características de padrões extraem propriedades globais da imagem tais como a área da forma, o centro de massa, o momento de inércia e assim por diante. A definição geral das funções de momento m_{pq} de ordem $(p+q)$ para uma uma função contínua de intensidade de imagem $X \times Y, f(x,y)$ é a que se segue:

$$m_{pq} = \int_y \int_x \psi_{pq}(x,y) f(x,y) dxdy \tag{2.1}$$

Onde p, q são inteiros entre $[0, \infty)$, x e y são as coordenadas de um ponto da imagem, e $\psi_{pq}(x,y)$ é a função de base. Propriedades valiosas, como a ortogonalidade, das funções de base são herdadas pelas funções de momento. Similarmente, a definição geral para uma imagem digital $X \times Y$ pode ser obtida substituindo as integrais por somatórias

$$m_{pq} = \sum_y \sum_x \psi_{pq}(x,y) f(x,y) \tag{2.2}$$

Onde p, q são inteiros entre $(0, \infty)$ que representam a ordem, x e y são as coordenadas de um ponto da imagem digital, e $\psi_{pq}(x,y)$ é a função de base. Uma propriedade desejável para qualquer característica de padrão é que a mesma seja invariante a translação, escala e rotação da imagem.

2.2.2.1 Momentos de Zernike

O processamento e análise de imagens se desenvolveu rapidamente desde 1960. Atualmente é amplamente usado em biomedicina, sensoriamento remoto, análise de documentos, etc. A seleção de boas características é um passo fundamental para a análise. Um sistema de reconhecimento flexível deve reconhecer a imagem independente de sua orientação, tamanho e localização no campo de visão, isto é, invariante à rotação, escala e translação. As propriedades invariantes dos momentos, formas 2D e 3D tem recebido considerável atenção nos últimos anos. Diferentes tipos de momentos que tem propriedades invariantes existem, como o momento geométrico, momento de Legendre, momento de Zernike, momento Pseudo-Zernike, momento rotacional, momento complexo. Para análise de imagens, três questões fundamentais relativas à sua utilidade devem ser consideradas. Eles incluem (1) sensibilidade ao ruído da imagem, (2) aspectos da redundância da informação e (3) capacidade de representação da imagem. Uma vantagem importante do uso de momentos para análise de imagens é que eles não são sensíveis ao ruído da imagem. Momentos de Zernike superam os outros no que diz respeito as características citadas, mas tem como desvantagem o fato que se as imagens são muito semelhantes, mas pertencentes a classes diferentes, precisamos calcular momentos de maior ordem, aumentando o custo computacional. Introduziremos os momentos de Zernike da seguinte

forma: Zernike introduziu um conjunto de polinômios complexos que formam um conjunto ortogonal completo no interior do círculo unitário $x^2 + y^2 = 1$. A forma desses polinômios é a que se segue.

$$\mathrm{V}_{nm}(\mathrm{x},\,\mathrm{y}) = \mathrm{V}_{nm}(\rho,\,\theta) = \mathrm{R}_{\mathrm{nm}}(\rho)\,\exp(\mathrm{jm}\theta),$$

onde n é um inteiro positivo ou zero, m números inteiros positivos e negativos sujeitos a restrições $n - |m|$ e $|m| \leq n$, ρ o comprimento do vetor da origem ao pixel (x, y), θ o ângulo entre o vetor ρ e o eixo x na direção horária, $R_{nm}(\rho)$ é o polinômio radial definido como

$$R_{nm}(\rho) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s[(n-s)!]\rho^{n-2s}}{s!(\frac{n+|m|}{2}-s)!(\frac{n-|m|}{2}-s)!}.$$

Observe que $R_{n,-m}(\rho) = R_{nm}(\rho)$, Esses polinômios são ortogonais e satisfazem

$$\int\int_{x^2+y^2\leq 1} [V_{nm}(x,y)]V_{pq}(x,y)dxdy = \tfrac{\pi}{n+1}\delta_{np}\delta_{mq},$$

$$\delta = \begin{cases} 1 & \text{se } (a = b) \\ 0 & \text{se } (a \neq b) \end{cases}$$

Os momentos de Zernike são a projeção da função da imagem nessas funções de base ortogonais. O momento Zernike de ordem n com repetição m para uma função de imagem contínua $f(x, y)$ que desaparece fora do círculo unitário é

$$A_{nm} = \frac{n+1}{\pi}\int\int_{x^2+y^2\leq 1} f(x,y)[V_{nm}(\rho,\theta)]dxdy.$$

Para uma imagem digital, as integrais são substituídas por somatórios como segue:

$$A_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x,y)V_{nm}(\rho,\theta),$$

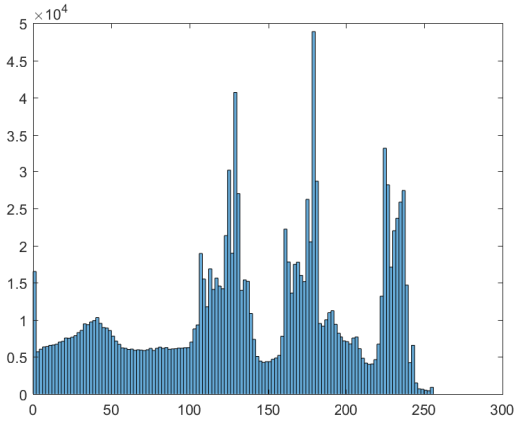
$$x^2 + y^2 \leq 1$$

Para calcular os momentos de Zernike de uma determinada imagem, o centro da imagem é tomado como origem e as coordenadas de pixel são mapeadas para o intervalo do círculo unitário. Os pixels que caem fora do círculo unitário não são usados na computação. Maiores detalhes sobre Momentos de Zernike pode ser encontrados em (ZHENJIANG, 2000).

2.2.3 Histograma

Segundo (PEDRINI; SCHWARTZ, 2008), o histograma de uma imagem corresponde à distribuição dos níveis de cinza da imagem, o qual pode ser representado por um gráfico indicando o número de pixels na imagem para cada nível de cinza. As figuras 6a e 6b mostram uma imagem e seu histograma correspondente. Seja $f(x, y)$ uma imagem

(a) Catedral Metropolitana de Maceió - Foto do autor



(b) Histograma de Níveis de Cinza.

Figura 6 – Imagem e seu respectivo histograma

representada por uma matriz bidimensional, com dimensões $M \times N$ pixels e contendo L níveis de cinza no intervalo $[0, L_{max}]$. O cálculo do histograma é apresentado na equação 2.3. O histograma é representado por um vetor H com L elementos. Uma imagem possui um único histograma, entretanto, a recíproca não é em geral verdadeira, pois um histograma não contém informação espacial, apenas valores de intensidade. O histograma pode ser visto como uma distribuição discreta de probabilidade, pois o número de pixels para um determinado nível de cinza pode ser utilizado para calcular a probabilidade de se encontrar um pixel com aquele valor de cinza na imagem. Dessa forma, o histograma $p_k(f)$ pode ser expresso como

$$p_k(f) = \frac{\eta_k}{\eta} = \frac{H(k)}{MN} \tag{2.3}$$

em que $\eta_k = H(k)$ representa o número de ocorrências do nível de cinza k e $\eta = MN$ corresponde ao número total de pixels da imagem f. Várias medidas estatísticas podem ser obtidas a partir do histograma de uma imagem, tais como os valores máximo e mínimo, o valor médio, a variância e o desvio padrão dos níveis de cinza da imagem.

2.2.4 Características de Direção

Caracteres compreendem cursos que são linhas orientadas, curvas ou linhas poligonais. A orientação ou direção desses cursos tem um papel importante na diferenciação entre vários caracteres. Por um longo tempo, a direção ou orientação dos cursos foi levada em conta no reconhecimento de caracteres baseado em análise de cursos. Para classificação estatística baseada no recurso de representação de vetor, caracteres também foram representados como estatísticas de vetores de orientação/direção. Para fazer isso, o ângulo de orientação/direção do curso é particionado num número fixo de faixas, e o número de segmentos de curso em cada faixa é tomado como um calor de característica. O conjunto dos números de segmentos orientacionais/direcionais forma assim um histograma,

chamado histograma de direção ou orientação. Para aumentar ainda mais a capacidade de diferenciação, o histograma é muitas vezes calculado para zonas locais da imagem de caracteres, dando o assim chamado histograma local de orientação / direção. Figura 7 mostra um exemplo de um histograma de orientação de contorno (quatro orientações, 4 x 4 zonas).

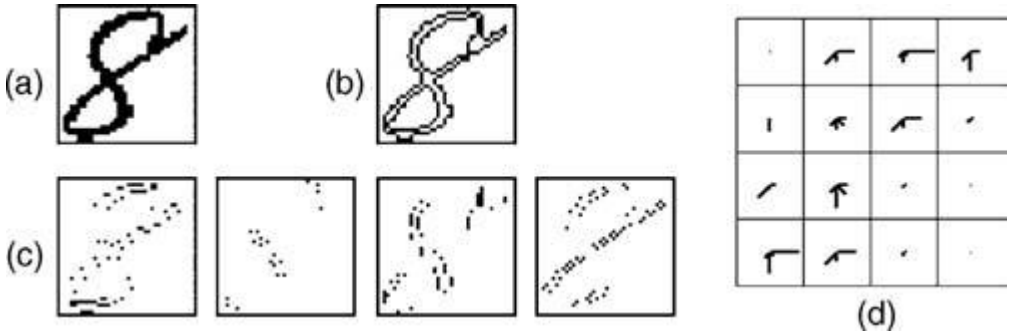


Figura 7 – (a) Caractere; (b) contorno de um caractere; planos de orientação; (d) histogramas de orientação locais.

2.2.5 Hierarchical Centroid - Centróide Hierárquico

A forma é um descritor de recurso significativo porque oferece outra maneira de descrever um objeto usando suas características mais importantes e diminuir o volume de informações armazenadas (ASIRI NORAH M.; ALHUMAIDI, 2015). A extração de características da forma é mais complexas do que a detecção, uma vez que a extração implica que temos uma descrição de uma forma, como sua posição e tamanho, enquanto a detecção de uma forma simplesmente implica conhecimento de sua existência dentro de uma imagem. Um método de extração de forma disponível deve extrair formas indefectivelmente, independentemente do valor de qualquer parâmetro que possa controlar a aparência de uma forma. Portanto, neste método de extração, buscamos propriedades de invariância que o processo de extração não varia de acordo com configurações e condições estabelecidas. O centróide hierárquico é um descritor rápido e simples onde um dos passos significativos neste método é a obtenção do centróide. Matematicamente, um centróide é o ponto central de uma área. O centróide de uma forma simples como um quadrado, triângulo ou círculo é fácil de detectar e simples de calcular. No entanto, o ponto central de uma forma irregular como no nosso conjunto de dados não é tão clara e o cálculo da sua localização pode ser difícil. Tais formas irregulares necessitam de um maior número de momentos para encontrar o centróide e esta freqüência de medição deve executar mais precisamente. Assim, o processo de obtenção do centróide inclui tomar a distância média em cada direção e expressar isso como uma proporção da área total de uma forma. Cada ponto na mudança de tamanho da forma é conhecido como um momento. A Figura 8 mostra o centróide da região da imagem onde f (x) e g (x) são curvas limitadas. Para calcular o centro desta

massa (forma irregular), temos primeiro a calcular a área desta massa como a seguir:

$$\mathrm{A} = \int_a^b [f(x) - g(x)]dx \tag{2.4}$$

O centróide de uma forma pode ser representado por $(\overline{x}, \overline{y})$ como segue

$$\overline{x} = \frac{1}{A}\int_a^b [x(f(x) - g(x)]dx \tag{2.5}$$

$$\overline{y} = \frac{1}{A}\int_a^b [\frac{1}{2}x(f(x)^2 - g(x)^2]dx \tag{2.6}$$

O método de extração de característica captura recursivamente as características de uma imagem pelo cálculo dos centróides das sub imagens retangulares hierarquicamente. A entrada do descritor é uma imagem bidimensional binária invertida e a saída é um vetor de níveis que é a profundidade dos locais de divisão.

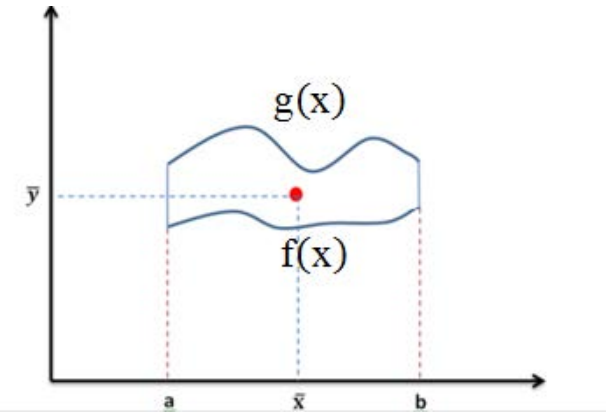


Figura 8 – Centróide de uma forma irregular

2.3 O Paradigma da Aprendizagem de Máquina

A aprendizagem de máquina (SUTHAHARAN, 2016) é sobre a exploração e o desenvolvimento de modelos e algoritmos matemáticos para aprender com os dados. Seu paradigma se concentra em objetivos de classificação e consiste em modelar um mapeamento ótimo entre o domínio de dados e o conjunto de conhecimento e desenvolver os algoritmos de aprendizagem. A classificação também é chamada de aprendizagem supervisionada, que requer um conjunto de dados de treinamento (rotulado), um conjunto de dados de validação e um conjunto de dados de teste. No entanto, para explicar brevemente, o conjunto de dados de treinamento ajuda a encontrar os parâmetros ótimos de um modelo, o conjunto de dados de validação ajuda a evitar a superposição do modelo e o conjunto de dados de teste ajuda a determinar a precisão do modelo.

2.3.1 Modelagem e Algoritmos

O termo modelagem refere-se a modelagem matemática e estatística de dados. O objetivo da modelagem é desenvolver um mapeamento parametrizado entre o domínio de dados e o conjunto de respostas. Este mapeamento pode ser uma função parametrizada ou um processo parametrizado que aprenda as características de um sistema a partir dos dados de entrada (rotulados). O termo algoritmo é um termo confuso no contexto da aprendizagem por máquinas. Para um cientista da computação, o termo algoritmo significa instruções sistemáticas passo-a-passo para um computador para resolver um problema. Na aprendizagem mecânica, a modelagem, em si, pode ter vários algoritmos para derivar um modelo; no entanto, o termo algoritmo aqui se refere a um algoritmo de aprendizagem. O algoritmo de aprendizagem é usado para treinar, validar e testar o modelo usando um dado conjunto de dados para encontrar um ótimo valor para os parâmetros, validá-lo e avaliar seu desempenho.

2.3.2 Supervisionada e não Supervisionada

É melhor definir aprendizagem supervisionada e aprendizagem não supervisionada com base na definição da classe. Na aprendizagem supervisionada, as classes são conhecidas e os limites das classes estão bem definidos no conjunto de dados (treinamento) fornecido, e a aprendizagem é feita usando essas classes (ou seja, etiquetas de classe). Por isso, é chamado de classificação. Na aprendizagem sem supervisão, assumimos que as classes ou os limites das classes não são conhecidos, portanto, os próprios rótulos de classe também são aprendidos, e as classes são definidas com base nisso. Portanto, os limites das classes são estatísticos e não definidamente definidos, e é chamado de clustering.

2.4 Ferramenta matemática usada para Classificação

2.4.1 Linguagem Python - Scikit-learn

Conforme (MüLLER, 2016), Python tornou-se a lingua franca para muitas aplicações da ciência de dados. Ele combina o poder de linguagens de programação de uso geral com a facilidade de uso de linguagens de script de domínio específicos como MATLAB ou R. Python tem bibliotecas para carregamento de dados, visualização, estatísticas, processamento de linguagem natural, processamento de imagens e muito mais. Esta vasta caixa de ferramentas fornece aos cientistas de dados uma grande variedade de funcionalidades para fins gerais e especiais. Uma das principais vantagens do uso do Python é a capacidade de interagir diretamente com o código, usando um terminal ou outras ferramentas. Aprendizagem de máquina e análise de dados são basicamente processos iterativos, nos quais os dados impulsionam a análise. É essencial que esses processos tenham ferramentas que permitam

iteração rápida e interação fácil. Como uma linguagem de programação de propósito geral, Python também permite a criação de interfaces gráficas de usuário complexas (GUIs) e serviços web, e para integração em sistemas existentes. Scikit-learn (PEDREGOSA et al., 2011) é um projeto de código aberto, o que significa que ele é livre para usar e distribuir, e qualquer um pode facilmente obter o código-fonte para ver o que está acontecendo nos bastidores. O projeto scikit-learn está constantemente sendo desenvolvido e aprimorado, e tem uma comunidade de usuários muito ativa. Ele contém uma série de algoritmos de aprendizado de máquina de alto nível, bem como documentação abrangente sobre cada algoritmo. Scikit-learn é uma ferramenta muito popular, é a biblioteca de Python mais proeminente para aprendizagem de máquina. É amplamente utilizado na indústria e na academia, e uma riqueza de tutoriais e trechos de código estão disponíveis on-line. Scikit-learn funciona bem também com uma série de outras ferramentas científicas Python, tais como NumPy e SciPy.

2.4.2 Algoritmos de Classificação

2.4.2.1 Algoritmo do Vizinho mais Próximo

O princípio por trás do Método dos Vizinhos mais Próximos (*Nearest Neighbors* - (MITCHELL, 1997)), é encontrar um número predefinido de amostras de treinamento mais próximas na distância ao novo ponto e prever o rótulo desses. O número de amostras pode ser uma constante definida pelo usuário (aprendendo o vizinho mais próximo), ou variar com base na densidade local dos pontos (aprendizado baseado no raio). A distância pode, em geral, ser qualquer medida métrica: a distância Euclidiana padrão é a escolha mais comum. Os métodos baseados em vizinhos são conhecidos como métodos de aprendizado de máquinas não generalizáveis, pois simplesmente "lembram"todos os seus dados de treinamento (possivelmente transformados em uma estrutura de indexação rápida, como *Ball Tree* ou *KD Tree*). Apesar da sua simplicidade, o método dos Vizinhos mais Próximos teve sucesso em uma grande quantidade de problemas de classificação e regressão, incluindo reconhecimento de dígitos manuscritos ou imagens de satélite. Sendo um método não paramétrico, muitas vezes é bem sucedido em situações de classificação onde o limite de decisão é muito irregular.

2.4.2.2 Máquina de Vetores de Suporte Linear e Máquina de Vetores de Suporte Linear com Funções de Base Radial - Linear SVM e RBF SVM

As máquinas de vetor de suporte (SVMs), (MITCHELL, 1997), são um conjunto de métodos de aprendizagem supervisionados utilizados para a classificação, regressão e detecção de valores atípicos. As vantagens das máquinas de vetor de suporte são:

- Eficaz em espaços dimensionais elevados.
- Ainda eficaz nos casos em que o número de dimensões é maior que o número de amostras.
- Usa um subconjunto de pontos de treinamento na função de decisão (chamados vetores de suporte), por isso também é eficiente em memória.
- Versátil: diferentes funções do Kernel podem ser especificadas para a função de decisão. São fornecidos kernels comuns, mas também é possível especificar kernels personalizados.

As desvantagens das máquinas de vetor de suporte incluem:

- Se o número de recursos for muito maior do que o número de amostras, evite o excesso de ajuste na escolha das funções do Kernel e o prazo de regularização é crucial.
- SVMs não fornecem provas de probabilidade diretamente, estas são calculadas usando uma avaliação cruzada de cinco vezes.

Uma máquina de vetor de suporte constrói um hiper-plano ou conjunto de hiper-planos em um espaço dimensional alto ou infinito (figura 9). Usamos (MERCER, 1909) para redução da dimensionalidade. Intuitivamente, uma boa separação é alcançada pelo hiper-plano que tem a maior distância para os pontos de treinamento mais próximos de qualquer classe (a chamada margem funcional), uma vez que, em geral, quanto maior a margem, menor será o erro de generalização do classificador. As máquina de suporte podem ser usadas para classificação, regressão ou outras tarefas.

2.4.2.3 Árvores de Decisão

Árvores de decisão (DTs - Decision Trees), (MITCHELL, 1997), são um método de aprendizagem supervisionado não paramétrico que pode ser usado para classificação e regressão. O objetivo é criar um modelo que preveja o valor de uma variável de destino aprendendo regras de decisão simples inferidas dos recursos de dados. Por exemplo, na figura 10 abaixo, as árvores de decisão aprendem com os dados para aproximar uma curva de seno com um conjunto de regras de decisão if-then-else. Quanto mais profunda a árvore, mais complexas são as regras de decisão e ajustar o modelo.

2.4.2.4 Floresta Aleatória

Uma floresta aleatória (*Random Forest*) é uma generalização inteligente de uma árvore de decisão (STAMP, 2017), figura 11. Uma árvore de decisão é uma estrutura de

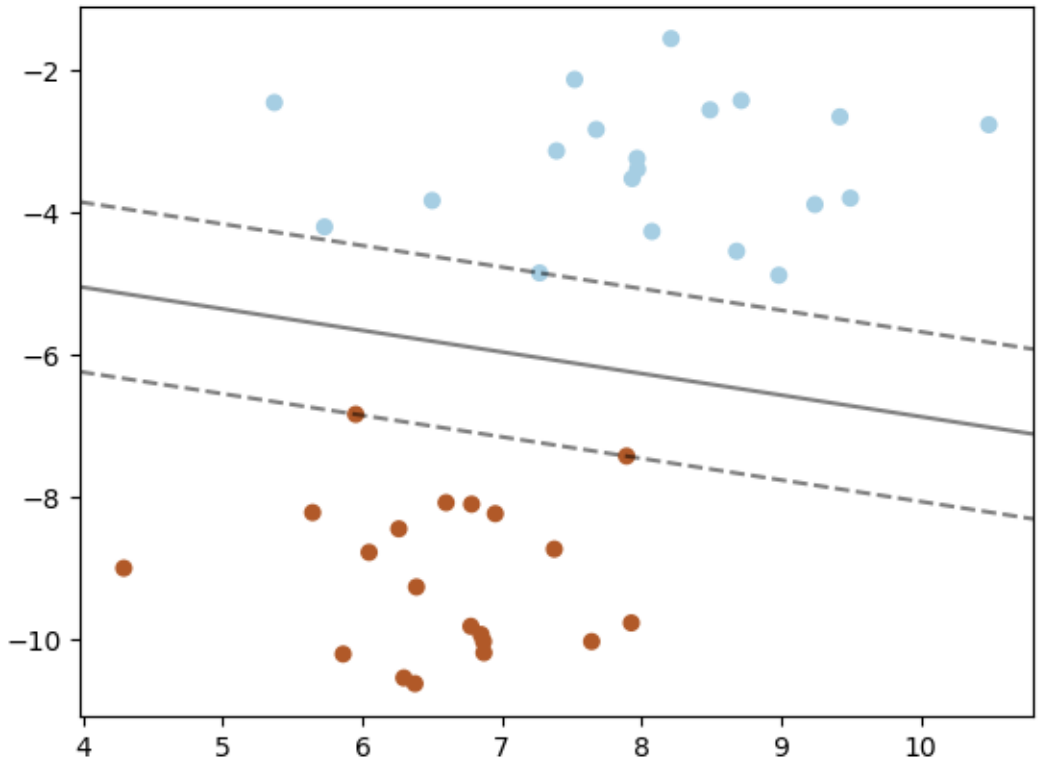


Figura 9 – SVC - Exemplo de Hiperplano Separador

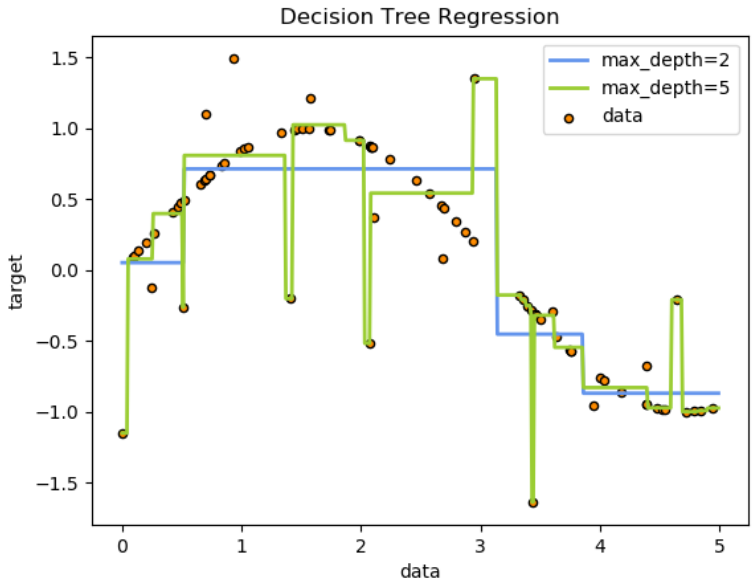


Figura 10 – Árvore de Decisão - Decision Tree

árvore simples que pode ser construída a partir de dados de treinamento rotulados.Uma árvore de decisão treinada pode ser usada para classificar dados de forma eficiente. Uma maneira de generalizar uma árvore de decisão é treinar várias árvores. Poderíamos selecionar subconjuntos diferentes (e sobrepostos) dos dados de treinamento e construir uma árvore para cada subconjunto, depois usar uma votação majoritária das árvores resultantes para determinar a classificação. Isso é conhecido como ensacar (*bagging*) as observações, e essa estratégia é menos propensa ao sobre-ajuste (*overfitting*), uma vez que tende a generalizar melhor os dados de treinamento. Em uma floresta aleatória, a idéia de ensacar é levada um passo adiante - além de ensacar as observações, também mantem os recursos. Ou seja, construímos árvores de decisão múltiplas a partir de seleções (com substituição) dos dados de treinamento, e também para várias seleções dos recursos (ou seja, vários subconjuntos e ordens dos recursos). Em seguida, para a classificação, combinamos o resultado de todas as árvores de decisão resultantes para obter a classificação final.

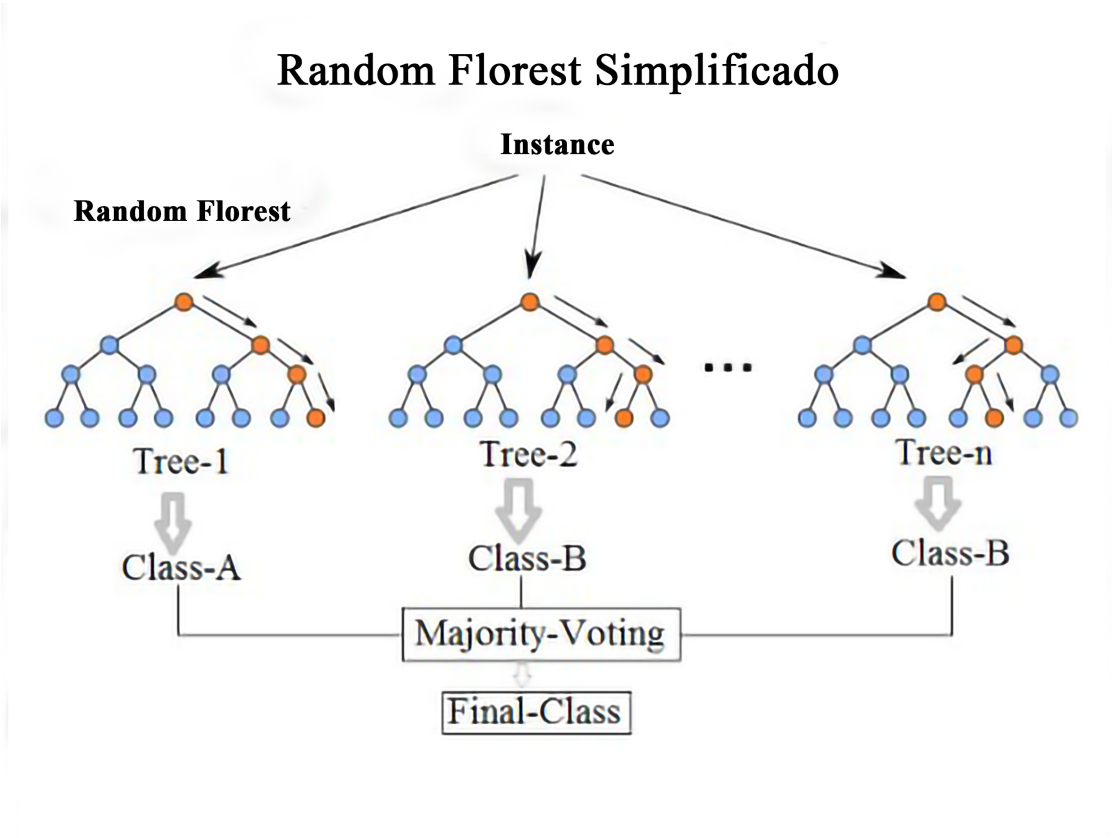


Figura 11 – Floresta Aleatória Simplificada

2.4.2.5 Rede Neural

Segundo (HAYKIN, 2008), uma rede neural (*neural network*) é um processador paralelo massivamente distribuído feito de unidades de processamento simples que tem a

propensão de armazenar conhecimento adquirido com a experiência e fazê-lo disponível para o uso. Ela assemelha-se ao cérebro em dois aspectos:

1. Conhecimento é adquirido pela rede do seu ambiente através de um processo de aprendizado.
2. Os pontos fortes da conexão entre neurônios, conhecidos como pesos sinápticos, são usados para armazenar o conhecimento adquirido.

O procedimento utilizado para realizar o processo de aprendizagem é chamado de algoritmo de aprendizado, cuja função é modificar os pesos sinápticos da rede de forma ordenada para atingir o objetivo de projeto desejado. A modificação dos pesos sinápticos fornece o método tradicional para o projeto de redes neurais. Também é possível que uma rede neural modificar sua própria topologia, que é motivado pelo fato de que os neurônios no cérebro humano podem morrer e novas conexões sinápticas podem crescer. Vemos na figura 12, um modelo simples de rede.

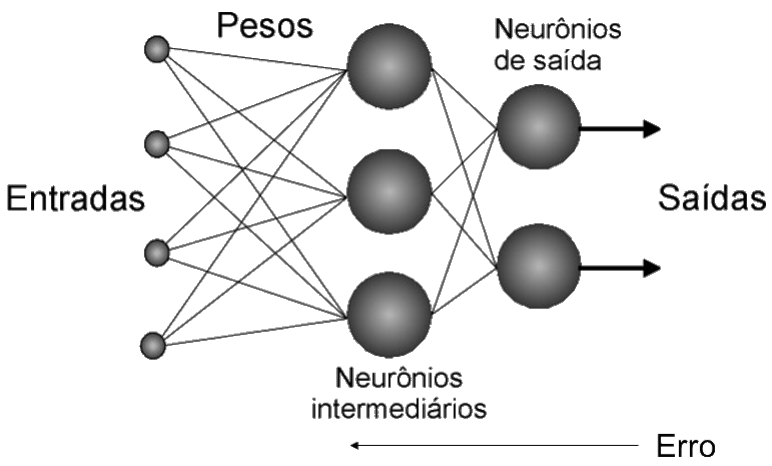


Figura 12 – Rede Neural

2.4.2.6 Adaboost

A técnica AdaBoost (abreviação de *Adaptive Boosting* - (HARRINGTON, 2012)) funciona de maneira ligeiramente diferente de muitos outros classificadores. A estrutura básica por trás dela pode ser uma árvore de decisão, mas o conjunto de dados usado para treinamento é continuamente adaptado para forçar o modelo a se concentrar nas amostras que estão mal classificadas. Além disso, os classificadores são adicionados sequencialmente, de modo que um novo impulsiona (*boosting*) o anterior, melhorando o desempenho nas áreas em que não era tão preciso quanto o esperado. Em cada iteração, um fator de peso é aplicado a cada amostra de modo a aumentar a importância das amostras que são

preditas erroneamente e diminuir a importância de outras. Em outras palavras, o modelo é repetidamente impulsionado, começando como um aprendiz muito fraco até que o número máximo de n estimadores seja atingido. As previsões, neste caso, sempre são obtidas por maioria de votos.

2.4.2.7 *Naive Bayes*

Naive Bayes - (HARRINGTON, 2012) é uma família de classificadores poderosos e fáceis de treinar que determinam a probabilidade de um resultado dado um conjunto de condições usando o teorema de Bayes. Em outras palavras,as probabilidades condicionais são invertidas, de modo que a consulta pode ser expressa como uma função de quantidades mensuráveis. A abordagem é simples, e o adjetivo "ingênuo"(*Naive*) foi atribuído não porque esses algoritmos sejam limitados ou menos eficientes, mas por causa de uma suposição sobre os fatores causais que vamos discutir. Naive Bayes são classificadores de múltiplo propósito e é fácil encontrar sua aplicação em muitos contextos diferentes; no entanto, seu desempenho é particularmente bom em todas as situações em que a probabilidade de uma classe é determinada pelas probabilidades de alguns fatores causais. Um bom exemplo é processamento de linguagem natural, onde um pedaço de texto pode ser considerado como uma instância específica de um dicionário e as frequências relativas de todos os termos fornecem informações suficientes para inferir uma classe ao qual ele pertença.

2.4.2.8 Análise Discriminante

A análise discriminante (DA - (SUTHAHARAN, 2016)) é uma técnica utilizada para desenvolver funções compostas pela combinação linear de variáveis independentes que são capazes de discernir entre diferentes categorias da variável dependente, ou seja, uma técnica multivariada utilizada quando a variável dependente é qualitativa (categórica) e as variáveis independentes são quantitativas (métricas). Isso nos permite examinar se existem diferenças significativas entre os grupos, em termos de variáveis preditoras. Ela também avalia a precisão da classificação. Testamos diferentes variações da análise discriminante; Análise de Discriminação Linear (LDA), Análise do Discriminante Quadrático (QDA) e Análise Discriminante Subsuperficial (SDA).

2.4.2.9 Conjunto

O conceito de conjunto (*ensemble* - (HARRINGTON, 2012) consiste em combinar vários modelos usando o mesmo algoritmo. Eles são um subconjunto de um conjunto mais amplo de técnicas denominadas multi-classificadores, onde múltiplos métodos de classificação são fundidos para resolver um problema único.

2.4.2.10 Regressão Logística

O objetivo da Regressão Logística (LR - *Logistic Regression* - (HARRINGTON, 2012)) é encontrar o modelo de melhor ajuste para descrever a relação entre a característica dicotômica de interesse e um conjunto de variáveis independentes (preditoras ou explicativas). A regressão logística gera os coeficientes (e seus erros padrão e níveis de significância) de uma fórmula para prever uma transformação logística da probabilidade de presença da característica de interesse. No nosso caso, as categorias são positivas (a imagem contém uma rasura) ou negativa (a imagem não contém uma rasura).

3 Proposta do Classificador

3.1 Metodologia de Trabalho

Uma visão geral da nossa proposta metodológica é a seguinte: Primeiro usaremos um banco de dados de imagens de rasuras e textos obtidos de trabalhos manuscritos de alunos do IFAL e testaremos características de imagens diversas individualmente e de todas as combinações possíveis. Em seguida, repetiremos todos os testes em um banco de dados similar, mas onde as imagens sem rasura e com rasura tem correlação, isto é, uma imagem rasurada é uma normal aonde foi aplicada a rasura. Por último tentaremos estender nossos processos para reconhecimento de dígitos, usando a base de dados MNIST. Teremos uma etapa de pré-processamento para normalização das figuras, primeiro com relação às dimensões, tornando-as de dimensões $n \times n$, n par, depois com relação à quantidade de cores, tornando a imagem binária (pontos totalmente brancos ou ou totalmente pretos), sempre que a função que extraia a característica só aceite imagens com formato ou quantidade de cores pré-determinados. Usaremos o Framework para Python Scikit-learn (PEDREGOSA et al., 2011). Foram usados os algoritmos de classificação *Nearest Neighbors*, *Linear SVM*, *RBF SVM*, *Decision Tree*, *Random Forest*, *Neural Network*, *Adaboost*, *Naive Bayes* e QDA (*Quadratic Discriminant Analysis*). Os classificadores utilizados nos gráficos serão os que apresentaram melhores resultados para efeitos de comparação.

3.1.1 Extração de Características

3.1.1.1 Característica 1 (EOH - *Edge Orientation Histogram* - Histograma de Orientação de Borda)

A ideia básica nesse passo é para construir um histograma com as direções dos gradientes dos cantos (bordas ou contornos). É possível detectar bordas em uma imagem mas nos interessa a detecção de ângulos. Isto é possível através de detectores de borda, tais como Canny, Prewitt, LoG, Roberts e Sobel . Os próximos cinco operadores na figura 14 podem dar uma ideia da força do gradiente, nessas cinco direções em particular. Para fazermos isso usamos um processo chamado convolução. A convolução é uma operação em duas funções f e g, que produz uma terceira função que pode ser interpretada como uma versão modificada ("filtrada") de f. Nesta interpretação, chamamos a função g de filtro. Se f é definido por uma variável espacial x ao invés de uma variável de tempo t, chamamos a operação de convolução espacial. A convolução é o princípio básico de qualquer dispositivo físico ou procedimento computacional que realize suavização ou aguçamento. Aplicado à funções bidimensionais como imagens, também é útil para descobertas de bordas, detecção

de características, detecção de movimento, correspondência de imagens e inúmeras outras tarefas. Formalmente, para as funções f (x) e g (x) de uma variável contínua x, a convolução é definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau).g(x-\tau)d\tau \tag{3.1}$$

Onde $*$ é a operação de convolução e . a operação de multiplicação ordinária. Para funções de uma variável discreta x, isto é, matrizes, a definição é:

$$f[x] * g[x] = \sum_{k=-\infty}^{\infty} f[k].g[x-k] \tag{3.2}$$

Finalmente, para funções de duas variáveis x e y (por exemplo, imagens), essas definições se tornam:

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2), g(x-\tau_1, y-\tau_2)d\tau_1 d\tau_2 \tag{3.3}$$

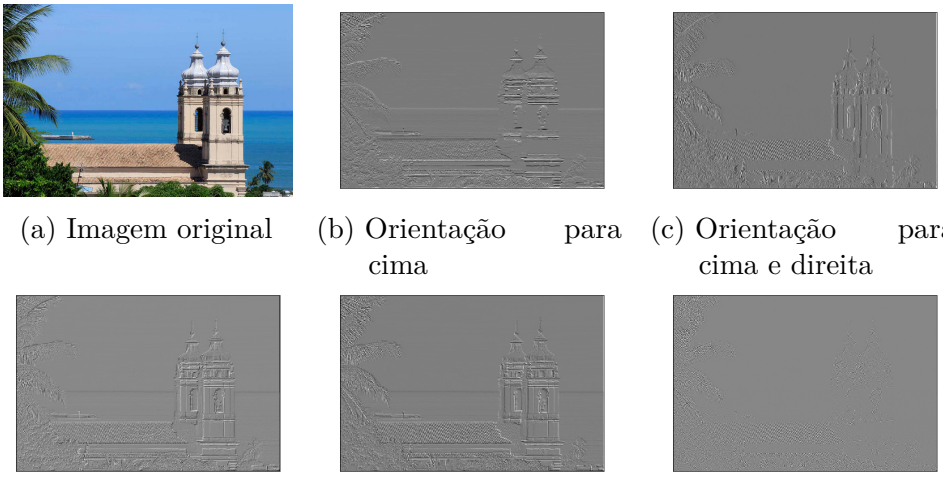
e

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2].g[x-n_1, y-n_2] \tag{3.4}$$

A convolução sobre cada uma dessas máscaras produz uma matriz do mesmo tamanho da imagem original indicando o gradiente (variação) da borda em uma direção particular. A figura 13 mostra o resultado de cada imagem quando convolucionada com cada filtro específico.

É possível contar o gradiente máximo nas 5 matrizes finais e usar isso para completar um histograma (Fig 15).

De maneira a evitar a quantidade de gradientes não importantes que poderiam potencialmente ser introduzidos por esta metodologia, uma opção é só levar em conta as bordas detectadas por métodos robustos como Canny, LoG, Prewitt, Roberts e Sobel. Estes detectores retornam uma matriz do mesmo tamanho da imagem,com um 1 se houver uma aresta e 0 se não houver a aresta. Basicamente ele retorna o contorno do objeto dentro da image. Considerando-se somente os 1's, estaremos contando os gradientes mais pronunciados. Como estamos interessados numa quantidade variável de características (cada histograma devolve 5 características de direção), dividiremos a imagem em partes de variam de 1 (a imagem completa) a 10 (100 sub-imagens), conforme ilustrado na figura 16.



(a) Imagem original (b) Orientação para cima (c) Orientação para cima e direita

(d) Orientação para direita (e) Orientação para baixo e direita (f) Sem orientação

Figura 13 – Imagens filtradas

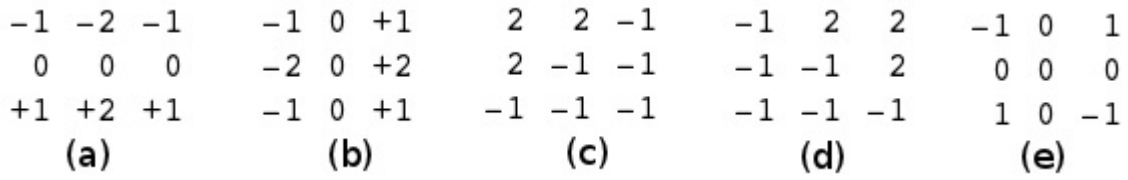


Figura 14 – As máscaras de sobel para 5 orientações: vertical, horizontal, diagonais e não-direcional

3.1.1.2 Característica 2 (HC - *Hierarchical Centroid* - Centróide Hierárquico)

Calculamos aqui os centróides hierárquicos de uma quantidade de sub-imagens, que varia conforme a profundidade da função, nos gerando vetores de características com comprimento de 4 (quatro) para um nível de profundidade de 2 até um máximo de 2044 para um nível de 10. A forma geral da quantidade de elementos do vetor de saída é $2 * (2^{profundidade} - 2)$. Na figura 17 vemos as subdivisões da imagem para o cálculo dos centróides hierárquicos.

3.1.1.3 Característica 3 (Momento de Zernike)

Usamos aqui uma função para o cálculo dos momentos de Zernike desenvolvida por (SAKI et al., 2013; TAHMASBI; SAKI; SHOKOUHI, 2011). Esta função encontra os momentos de Zernike para uma ROI (região de interesse) binária de n x n, onde N é um número par. Ela nos retorna um vetor de componentes [Z, A, Phi], onde Z é o momento complexo de Zernike, A a amplitude do momento e Phi a fase do ângulo do momento em graus. Os parâmetros aqui variados para utilização no classificador foram a ordem do momento Zernike (n) e o número de repetições do momento de Zernike (m). A figura 18 mostra a invariância do momento de Zernike à rotação. Vemos que os valores da amplitude

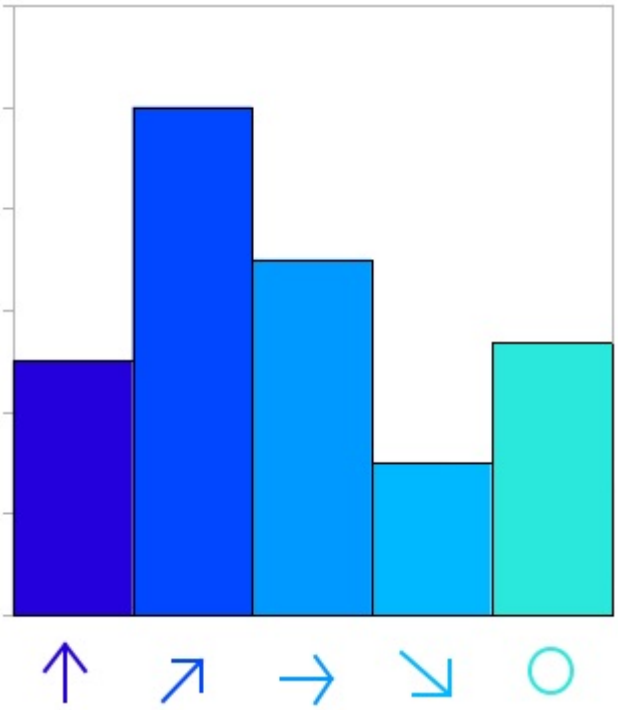
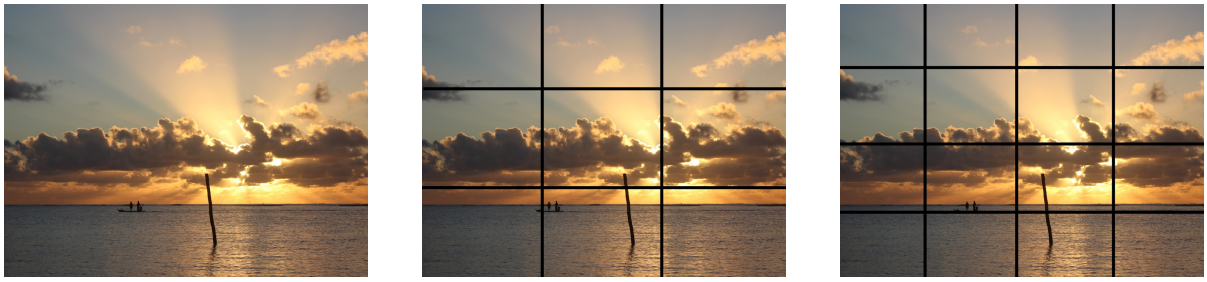


Figura 15 – Histograma de Orientação de Borda



(a) Divisão em regiões 1 X 1 (b) Divisão em regiões 3 X 3 (c) Divisão em regiões 8 X 8

Figura 16 – Subdivisões da Imagem - Foto do autor

(a) Amostra Negativa (prof.=2) (b) Amostra Positiva (prof.=2) (c) Amostra Negativa (prof.=7) (d) Amostra Positiva (prof.=2)

Figura 17 – Centróides Helicoidais em níveis de profundidade 2 e 7

do momento Complexo de Zernike variam muito pouco quando a figura está rotacionada.

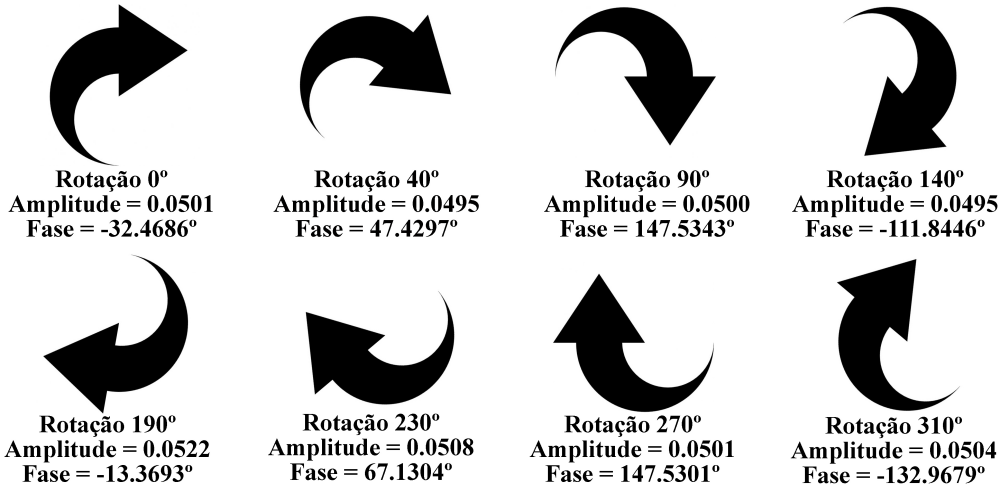


Figura 18 – Momento de Zernike para n=4 e m=2

3.1.2 Banco de Imagens

3.1.2.1 Base de Dados 1 (BD1)

A Base de Dados 1 consiste em 400 imagens de palavras e/ou letras. Destas, 200 contém rasuras (classe +1) enquanto 200 não possuem rasuras (classe -1), como ilustrado na Figura 19.

Esta base de dados inicial, foi obtida a partir do escaneamento de trabalhos efetuados por alunos do IFAL - Campus Viçosa, provas escritas e relatórios de trabalhos, durante aulas presenciais ministradas para o curso Técnico Subsequente de Informática. Essas rasuras são capturas de letras e/ou palavras completas e foram escolhidas aleatoriamente dentre os mais diversos tipos de rasuras presentes nos documentos (Figura 20).

3.1.2.2 Base de Dados 2 (BD2)

A Base de Dados 2 também consiste em 400 imagens de palavras e/ou letras. Destas, 200 contém rasuras (classe +1) enquanto 200 não possuem rasuras (classe -1), como ilustrado na Figura 21.

Esta segunda base de dados, foi também obtida a partir do escaneamento de trabalhos efetuados por alunos do IFAL - Campus Viçosa, durante aulas presenciais ministradas para o curso Técnico Subsequente de Informática. Essas rasuras são relacionadas, isto é, para cada uma imagem sem rasura, existe a mesma rasurada. São capturas de letras e/ou palavras completas e os caracteres e palavras rasuradas equivalentes foram gerados artificialmente por 3 (três) elementos humanos distintos. A letra e/ou palavra sem rasura, recebeu um rasura do tipo sobreposição ou apagamento para gerar o seu equivalente rasurado. Usamos pessoas distintas para garantir uma heterogeneidade na construção das rasuras. Os documentos utilizados foram os mesmos para geração de BD1, porém os

Figura 19 – Exemplos de imagens sem rasura (classe -1, coluna esquerda) e com rasura (classe +1, coluna direita) presentes na Base de Dados 1 (BD1).

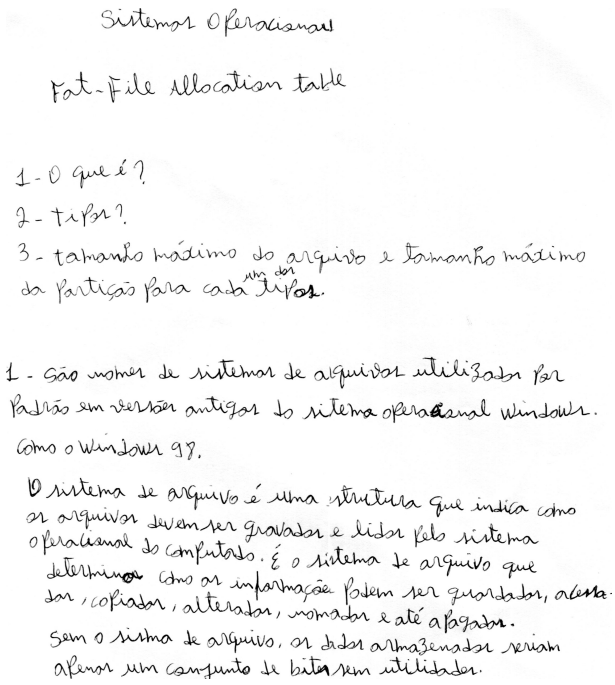


Figura 20 – Exemplo de documento escaneado com rasuras

caracteres e palavras selecionados não possuíam rasuras, sendo estas criadas artificialmente, para podermos ter uma equivalência entre caractere e palavras com e sem rasuras.

3.1.2.3 Base de Dados 3 (BD3)

A Base de dados 3 (BD3), MNIST ("*Mixed National Institute of Standards and Technology*- Misturada do Instituto Nacional de Padrões e Tecnologia (LECUN Y.; BOTTOU, 1998), foi construída a partir das bases do *NIST Special Database 3 (SD-3) e Special*

Figura 21 – Exemplos de imagens sem rasura (classe -1, coluna esquerda) e com rasura (classe +1, coluna direita) presentes na Base de Dados 2 (BD2).

Database 1 (SD-1), contendo imagens binárias de dígitos manuscritos. O Instituto Nacional de Padrões e Tecnologia - NIST, originalmente designou a base SD-3 como conjunto de treinamento e a base SD-1 como conjunto de teste. Contudo, SD-3 é muito mais limpa e fácil de reconhecer que SD-1. A razão para isto se deve ao fato que SD-3 foi coletada com os funcionários do Escritório do Censo norte-americano, enquanto SD-1 foi coletada com estudantes do ensino médio. Obter conclusões sensatas a partir de experiências de aprendizagem exige que o resultado seja independente da escolha do conjunto de treino e teste entre o conjunto completo de amostras. Portanto, foi necessário construir um novo banco de dados misturando os conjuntos de dados do NIST. SD-1 contém 58.527 imagens digitadas por 500 escritores diferentes. Em contraste com SD-3, onde blocos de dados de cada escritor aparecem em sequência, os dados em SD-1 são codificado. As identidades do escritor para SD-1 estão disponíveis e nós usamos essas informações para desembaralhar os escritores. Em seguida, dividimos SD-1 em dois: caracteres escritos pelos primeiros 250 escritores entraram em nosso novo conjunto de treinamento. Os restantes 250 escritores foram colocados em nosso conjunto de teste. Assim nós tivemos dois conjuntos com quase 30.000 exemplos cada. O novo conjunto de treinamento foi concluído com exemplos suficientes de SD-3, começando no padrão #0, para fazer um conjunto completo de 60.000 padrões de treinamento. De forma semelhante, o novo conjunto de teste foi completado com exemplos de SD-3 a partir do padrão #35.000 para fazer um conjunto completo com 60.000 padrões de teste. Nas experiências aqui descritas, utilizamos apenas um subconjunto de 10 000 imagens de teste (5.000 de SD-1 e 5.000 de SD-3).

As bases de dados estão sumarizadas na Tabela 1 abaixo.

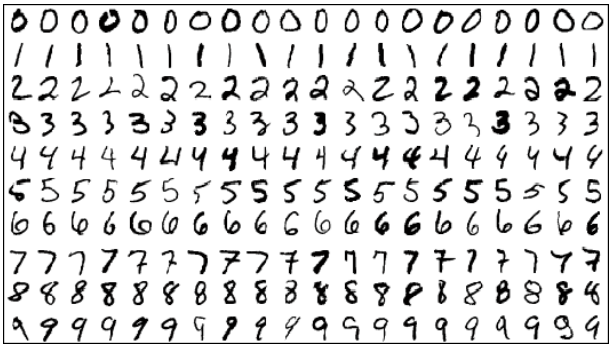


Figura 22 – Exemplos de dígitos da MNIST presentes na Base de Dados 3 (BD3).

Tabela 1 – Sumário dos Bancos de Dados.

Base de Dados	Quantidade de Amostras	Classes
BD1	400	2
BD2	400	2
BD3	10000	10

Fonte – Produzido pelo autor.

3.2 Algoritmos de Classificação

3.2.1 Preparação dos Dados

O software de processamento de imagem recebe um escrito escaneado como entrada, então segmenta cada palavra e as transforma em imagens no formato de matriz (bitmap). Textos com rasuras e textos normais foram escaneados e separados em duas classes, provendo o conhecimento ao sistema para identificação de outras amostras. As imagens de tamanho retangular, para serem processadas por funções que exigiam uma matriz $m \times m$ (quadrada), foram escalonadas para um tamanho de 200×200 pixels, sendo todas também, transformadas para escala de cinza de cores (256 níveis). Os dados do banco de dados 3 (BD3 - MNIST) foram utilizados sem nenhum tratamento, já que os mesmos estão num formato de matriz quadrada e em escala de cinza. Todos os métodos de aprendizagem supervisionada começam com uma matriz de dados de entrada X, resultado das funções de características aplicadas (Histograma de Orientação de Borda - EoH, Centróide Hierárquico - HC, Momentos de Zernike - ZM, e as combinações entre elas, EoH + HC, EoH + ZM, HC+ZM, EoH+HC+ZM), onde cada coluna de X representa uma característica, ou preditor. Pode-se usar qualquer tipo de dados para a matriz de resposta de dados Y, no nosso caso, usamos 1 (um) para representar uma imagem com rasura e -1 (menos 1) para representar uma imagem sem rasura. Para a maioria dos casos, não foi utilizado PCA para a redução de características.

3.2.2 Algoritmos

Há vantagens e desvantagens entre diversas características de algoritmos, tais como:

- Velocidade de treinamento
- Uso de memória
- Acurácia na predição de novos dados
- Transparência ou interpretabilidade, o que significa que facilmente você pode entender as razões pelas quais um algoritmo faz suas previsões.

Os algoritmos utilizados foram:

3.2.2.1 Algoritmos dos Vizinhos mais Próximos

Foram feitas duas variações nos parâmetros:

1. Número de vizinhos (1, 10 ou 100 - *Fine*, *Medium* ou *Coarse* respectivamente)
2. Forma de calcular a distância (métrica cosseno, métrica cúbica e ponderada).

3.2.2.2 Maquina de Vetor de Suporte

Foram ajustados os seguintes parâmetros:

1. Função do Kernel
   - Kernel Linear, mais fácil de ser interpretado
   - Gaussiano ou Kernel Função de Base Radial (RBF)
   - Quadratic
   - Cubic
2. Nível de Restrição da Caixa (*Box constraint level*) - a variação desse parâmetro limita os valores permitidos dos multiplicadores Lagrange. Para ajustar seu classificador de SVM, tente aumentar o nível de restrição de caixa. Aumentando esse nível pode-se diminuir o número de vetores de suporte, porém pode-se aumentar também o tempo de treinamento.
3. Modo Escala do Kernel *(Kernel Scale Mode)* - Especifica a escala manual do kernel se desejado. Quando configurado o modo de escala Kernel como Auto, o software usa um procedimento heurístico para selecionar o valor da escala. O procedimento heurístico usa subamostragem. Portanto, para reproduzir resultados, defina uma semente de número aleatório antes de treinar o classificador. Quando configurado o modo de escala do Kernel em Manual, pode-se especificar um valor.

3.2.2.3 Árvore de Decisão - *Decision Tree*

Variamos a quantidade de divisões do algoritmo (ramos da árvores) para máximos de 100, 20 e 4 (*Tree Fine, Tree Medium e Tree Coarse*). O critério de separação foi o índice de diversidade de Gini ((BREIMAN JEROME FRIEDMAN, 1984). Todas as características encontradas foram usadas (exceto quando existiam dados NaN (sem representação númerica), que eram automaticamente descatados. A redução de características (PCA) não foi utilizada, já que o procedimento adotado para geração de características, previa a geração de conjuntos com tamanhos diferentes.

3.2.2.4 Análise Discriminante

A análise discriminante é um popular primeiro algoritmo de classificação a ser tentado porque é rápido e fácil de interpretar. A análise discriminante é boa para conjuntos de dados com grande quantidade de elementos. A análise discriminante pressupõe que diferentes classes geram dados com base em diferentes distribuições gaussianas. Para treinar um classificador, a função de ajuste estima os parâmetros (média e covariância) de uma distribuição gaussiana para cada classe. Foram usados dois classificadores deste tipo: O Discriminante Linear e o Discriminante Quadrático.

3.2.2.5 Métodos de Conjunto (*Ensemble*)

Os métodos de conjunto utilizados foram Adaboost com Árvores de Decisão, Floresta Randômica com Árvores de Decisão, Subespacial com Discriminante e Vizinhos mais próximos. A flexibilidade do modelo aumenta com o número de aprendizes. Os classificadores de conjunto tendem a ser lentos, porque muitas vezes precisam de muitos aprendizes.

3.2.2.6 Outros

Os algoritmos não citados foram utilizados com sua configuração padrão, já que a ideia era tentar provar que a utilização de um conjunto de características num classificador, mesmo utilizado na sua forma mais simples (padrão), poderiam resultar em resultados melhores que apenas a utilização de cada uma delas isoladamente. Estudos mais detalhados de ajuste de cada um dos algoritmos e como estes ajustes podem ser feito em consideração à cada uma das características, ficam para um trabalho futuro.

4 Resultados e Discussões

4.1 Resultados Obtidos para o Banco de Dados 1 + Scikit-learn

4.1.1 Utilizando a característica 1 - Histograma de Orientação de Borda

Os parâmetros modificados aqui para uma melhoria na acurácia dos classificadores foram a quantidade de sub-imagens, variando de 1 até 100, e o tipo do filtro utilizado na função de cálculo do histograma (Canny, LoG, Prewitt, Roberts e Sobel). As tabelas mostram a variação da acurácia em função do quantidade de sub-imagens (melhores classificadores) e tipo de filtro (para uma mesma quantidade de sub-imagens).O melhor classificador geral foi utilizado para um comparativo entre os tipos de filtros. Apesar de não ser necessário para o cálculo do histograma, as imagens foram normalizadas para uma dimensão de 200×200 pixels, igualando-as às utilizadas no cálculo das outras características. As tabelas 2, 3, 4, 5 e 6 mostram esses resultados. O melhor valor obtido foi com o filtro LoG (*Log of Gaussian*) de 93,3%, e utilizaremos esse filtro para mostrar a variação em função da quantidade de sub-imagens, figura 23.

4.1.1.1 Filtro Canny

A utilização do Filtro Canny (tabela 2) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,94±0,03**, utilizando-se o classificador Processo Gaussiano.

Tabela 2 – Variação da Acurácia do Classificador em função da quantidade de sub-imagens utilizadas, utilizando-se o filtro Canny. Os valores exibidos na primeira coluna, formato S:C, representam a quantidade de cortes horizontais e verticais feitos na figura (S) e a quantidade de características geradas pelas sub-imagens (C)

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	0,87±0,05	0,89±0,06	0,88±0,05	0,90±0,07	**0,86±0,04**	**0,90±0,05**	0,89±0,07	0,88±0,08	**0,86±0,08**	**0,88±0,08**
2-20	**0,89±0,06**	0,91±0,03	**0,90±0,06**	0,93±0,05	0,81±0,09	0,89±0,11	0,90±0,03	0,89±0,03	0,82±0,09	0,87±0,09
3-45	0,82±0,10	0,91±0,07	0,88±0,08	0,91±0,07	0,83±0,06	0,83±0,11	0,91±0,06	0,87±0,11	0,84±0,09	0,84±0,12
4-80	0,80±0,08	0,91±0,04	0,87±0,08	0,93±0,03	0,79±0,04	0,84±0,07	0,91±0,06	**0,90±0,05**	0,84±0,10	0,79±0,13
5-125	0,77±0,12	0,90±0,04	0,86±0,05	0,93±0,03	0,75±0,14	0,87±0,12	0,92±0,03	0,88±0,06	0,82±0,11	0,68±0,08
6-180	0,72±0,13	0,91±0,07	0,85±0,05	**0,94±0,03**	0,74±0,11	0,82±0,06	**0,93±0,06**	0,89±0,06	0,81±0,14	0,56±0,11
7-245	0,69±0,09	0,91±0,03	0,82±0,06	**0,94±0,03**	0,71±0,09	0,86±0,14	0,91±0,06	0,86±0,07	0,79±0,11	0,53±0,08
8-320	0,67±0,14	**0,92±0,08**	0,79±0,05	0,94±0,04	0,73±0,08	0,82±0,08	0,91±0,09	0,86±0,11	0,75±0,18	0,55±0,08
9-405	0,62±0,12	0,90±0,06	0,75±0,07	0,93±0,03	0,68±0,06	0,81±0,09	0,91±0,06	0,89±0,06	0,75±0,14	0,52±0,11
10-500	0,60±0,12	0,87±0,05	0,71±0,07	0,58±0,31	0,74±0,06	0,82±0,08	0,87±0,08	0,87±0,05	0,75±0,15	0,52±0,08

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

4.1.1.2 Filtro LoG - Laplacian of Gaussian

A utilização do Filtro LoG (tabela 3) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,94±0,03**, utilizando-se o classificador Processo Gaussiano.

Tabela 3 – Variação da Acurácia com a característica 1 (EoH) utilizando-se Filtro LoG.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,93±0,09**	0,92±0,08	0,90±0,05	0,95±0,07	0,86±0,04	**0,91±0,10**	0,92±0,09	0,89±0,10	**0,88±0,09**	**0,92±0,06**
2-20	0,90±0,04	0,93±0,07	**0,91±0,05**	0,95±0,05	0,84±0,06	0,89±0,07	0,93±0,06	0,90±0,08	0,85±0,11	0,89±0,05
3-45	0,85±0,09	0,93±0,07	0,88±0,07	0,94±0,05	**0,87±0,03**	0,87±0,06	0,93±0,07	0,89±0,08	0,85±0,11	0,85±0,08
4-80	0,83±0,08	0,93±0,05	0,86±0,11	0,94±0,05	0,80±0,08	0,88±0,07	0,92±0,06	0,90±0,03	0,83±0,12	0,77±0,16
5-125	0,80±0,12	0,92±0,06	0,85±0,10	0,94±0,05	0,82±0,09	0,87±0,07	0,94±0,04	**0,92±0,05**	0,81±0,14	0,70±0,12
6-180	0,73±0,04	**0,93±0,03**	0,82±0,06	0,95±0,04	0,79±0,09	0,85±0,06	0,93±0,05	0,90±0,04	0,80±0,14	0,53±0,07
7-245	0,70±0,05	**0,93±0,03**	0,79±0,09	**0,96±0,03**	0,74±0,08	0,86±0,04	**0,95±0,04**	0,90±0,08	0,78±0,12	0,53±0,09
8-320	0,69±0,10	0,91±0,05	0,77±0,06	0,95±0,05	0,75±0,14	0,87±0,06	0,91±0,07	0,90±0,10	0,77±0,20	0,51±0,12
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Fonte: Autor

4.1.1.3 Filtro Prewitt

A utilização do Filtro Prewitt (tabela 4) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,79±0,07**, utilizando-se o classificador Rede Neural e 320 valores de orientação.

Tabela 4 – Variação da Acurácia com a característica 1 (EoH) utilizando-se Filtro Prewitt.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	0,71±0,17	0,76±0,20	**0,72±0,15**	0,77±0,16	**0,71±0,09**	**0,73±0,10**	0,76±0,21	**0,76±0,12**	0,73±0,12	**0,77±0,07**
2-20	0,70±0,12	0,78±0,16	0,72±0,16	**0,79±0,11**	0,68±0,07	0,70±0,07	0,75±0,15	0,71±0,09	0,71±0,11	0,72±0,04
3-45	**0,71±0,11**	**0,79±0,14**	0,70±0,13	**0,79±0,11**	0,66±0,05	0,70±0,12	0,76±0,14	0,74±0,09	0,72±0,06	0,72±0,05
4-80	0,70±0,10	0,78±0,15	0,70±0,14	0,77±0,15	0,62±0,08	0,67±0,09	0,76±0,16	0,72±0,11	**0,73±0,09**	0,69±0,10
5-125	0,68±0,07	0,76±0,09	0,70±0,14	0,78±0,11	0,59±0,11	0,70±0,10	0,78±0,10	0,72±0,07	0,72±0,09	0,65±0,14
6-180	0,66±0,13	0,77±0,10	0,69±0,13	0,78±0,10	0,64±0,08	0,69±0,07	0,78±0,07	0,72±0,09	0,70±0,09	0,53±0,08
7-245	0,66±0,09	0,77±0,13	0,66±0,13	0,77±0,14	0,60±0,10	0,68±0,07	0,78±0,07	0,72±0,07	0,69±0,07	0,51±0,15
8-320	0,62±0,10	0,78±0,07	0,63±0,10	0,78±0,11	0,63±0,11	0,70±0,10	**0,79±0,07**	0,71±0,05	0,68±0,07	0,57±0,13
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Fonte: Autor

4.1.1.4 Filtro Roberts

A utilização do Filtro Roberts (tabela 5) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,74±0,09**, utilizando-se o classificador Processo Gaussiano e 45 valores de orientação.

Tabela 5 – Variação da Acurácia com a característica 1 (EoH) utilizando-se Filtro Roberts.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	0,63±0,08	0,66±0,11	**0,64±0,09**	0,71±0,05	**0,66±0,06**	0,68±0,14	0,64±0,07	0,69±0,11	0,66±0,13	**0,70±0,20**
2-20	0,60±0,10	0,65±0,08	0,62±0,08	0,69±0,09	0,58±0,08	0,62±0,09	0,63±0,09	0,66±0,06	0,65±0,12	0,68±0,10
3-45	**0,65±0,12**	**0,73±0,09**	0,64±0,10	**0,74±0,09**	0,66±0,08	0,66±0,10	0,68±0,13	0,67±0,08	0,65±0,10	0,68±0,10
4-80	0,62±0,13	0,71±0,12	0,64±0,10	0,73±0,09	0,60±0,07	**0,70±0,09**	0,71±0,17	**0,73±0,07**	**0,67±0,14**	0,64±0,06
5-125	0,63±0,09	0,73±0,04	0,63±0,09	0,73±0,09	0,57±0,03	0,66±0,16	**0,72±0,07**	0,68±0,08	0,65±0,10	0,58±0,05
6-180	0,58±0,10	0,67±0,04	0,62±0,07	0,71±0,09	0,60±0,09	0,66±0,06	0,68±0,09	0,71±0,09	0,64±0,12	0,52±0,06
7-245	0,60±0,07	0,66±0,08	0,62±0,06	0,67±0,13	0,61±0,06	0,63±0,10	0,68±0,12	0,72±0,11	0,61±0,12	0,53±0,04
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Fonte: Autor

4.1.1.5 Filtro Sobel

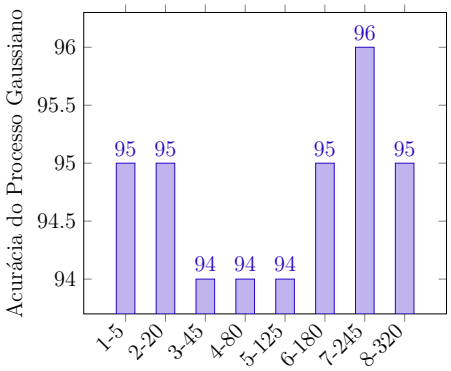
A utilização do Filtro Sobel (tabela 6) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,82±0,05**, utilizando-se o Processo Gaussiano e 180 valores de orientação.

Tabela 6 – Variação da Acurácia com a característica 1 (EoH) utilizando-se Filtro Sobel.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	0,72±0,17	0,76±0,21	**0,73±0,15**	0,80±0,08	**0,69±0,16**	0,72±0,18	0,76±0,21	**0,76±0,11**	**0,73±0,12**	**0,77±0,10**
2-20	0,70±0,13	0,78±0,16	0,71±0,15	0,79±0,07	0,67±0,12	0,70±0,07	0,75±0,18	0,73±0,10	0,71±0,11	0,72±0,04
3-45	**0,72±0,08**	**0,79±0,14**	0,70±0,13	0,80±0,11	0,65±0,13	0,68±0,12	0,77±0,13	0,72±0,05	0,72±0,06	0,73±0,04
4-80	0,70±0,10	0,77±0,13	0,69±0,14	0,80±0,07	0,60±0,13	0,70±0,08	0,76±0,15	**0,76±0,11**	0,72±0,08	0,70±0,07
5-125	0,68±0,04	0,77±0,08	0,71±0,14	0,80±0,04	0,62±0,09	**0,72±0,07**	0,77±0,10	0,74±0,06	0,72±0,09	0,65±0,13
6-180	0,65±0,13	0,77±0,10	0,69±0,13	**0,82±0,05**	0,64±0,04	0,69±0,04	0,77±0,08	0,73±0,03	0,70±0,09	0,53±0,12
7-245	0,66±0,07	0,77±0,11	0,66±0,12	0,78±0,09	0,59±0,07	0,70±0,07	0,77±0,08	0,73±0,11	0,70±0,08	0,52±0,17
8-320	0,64±0,09	0,77±0,05	0,63±0,10	0,79±0,08	0,61±0,11	0,66±0,09	**0,78±0,06**	0,71±0,08	0,67±0,07	0,54±0,07
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Fonte: Autor

Figura 23 – Filtro LoG X Classificador X Sub-Imagens



4.1.2 Utilizando a característica 2 - Centróide Hierárquico

Variaremos aqui a quantidade de sub-imagens utilizadas para o cálculo das centróides (tabela 7). O valor máximo obtido foi de **0,92±0,04** para os classificadores Adaboost com 12 características e *Naive Bayes* com 60 características.

Tabela 7 – Resultados do Hierarchical Centroid. A coluna Profundidade-Características representa a quantidade de sub-imagens e a quantidade de características geradas por ela.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-4	0,69±0,06	0,49±0,09	0,49±0,09	0,73±0,07	0,65±0,07	0,65±0,05	0,56±0,13	0,70±0,05	0,70±0,07	0,73±0,06
3-12	0,89±0,04	0,89±0,05	0,88±0,05	0,91±0,05	**0,89±0,05**	0,91±0,05	0,88±0,05	**0,92±0,04**	0,88±0,05	**0,86±0,04**
4-28	0,90±0,06	0,89±0,06	0,88±0,05	**0,92±0,05**	**0,89±0,05**	0,90±0,06	0,89±0,06	0,91±0,05	0,90±0,05	0,83±0,04
5-60	0,90±0,06	**0,89±0,04**	0,87±0,05	0,91±0,04	0,85±0,06	**0,91±0,04**	**0,89±0,05**	0,90±0,02	**0,92±0,04**	0,77±0,06
6-124	**0,90±0,05**	0,88±0,05	**0,88±0,04**	0,91±0,04	0,84±0,05	0,89±0,04	**0,89±0,05**	0,91±0,05	0,91±0,03	0,60±0,17
7-252	0,89±0,05	0,88±0,05	0,87±0,05	0,91±0,04	0,85±0,07	0,90±0,05	**0,89±0,05**	0,89±0,05	0,91±0,04	0,50±0,08
8-508	0,89±0,05	0,87±0,04	0,87±0,04	0,91±0,04	0,84±0,05	0,90±0,04	0,88±0,06	0,89±0,05	0,91±0,04	0,50±0,11
9-1020	0,89±0,05	0,86±0,04	0,87±0,04	0,91±0,04	0,84±0,06	0,89±0,05	0,89±0,06	0,90±0,04	0,91±0,04	0,50±0,09
10-2044	0,89±0,05	0,86±0,05	0,87±0,06	0,91±0,03	0,84±0,04	0,90±0,03	0,88±0,06	0,90±0,05	0,91±0,04	0,51±0,09

Fonte: Autor

4.1.3 Utilizando a característica 3 - Momentos de Zernike

Para esta característica analisaremos dois casos:

1. variando a ordem do polinômio (2-16, ordem par - tabela 8)
2. variando a quantidade de repetições do Momento de Zernike (2-4, ordem par - tabela 9)

Os valores máximos obtidos foram **0,71±0,12** para Discriminante Quadrático em ambos os casos.

Tabela 8 – Valores para o Momento de Zernike variando a ordem.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-2	**0,66±0,10**	0,69±0,07	0,66±0,08	0,66±0,13	0,57±0,10	0,60±0,08	0,64±0,08	0,60±0,10	0,68±0,16	0,64±0,12
4-2	0,64±0,11	**0,71±0,17**	**0,71±0,14**	**0,70±0,14**	**0,60±0,09**	**0,63±0,08**	**0,70±0,16**	**0,67±0,09**	**0,71±0,15**	**0,71±0,12**
6-2	0,53±0,11	0,57±0,12	0,56±0,10	0,56±0,09	0,52±0,07	0,56±0,05	0,56±0,11	0,50±0,10	0,57±0,12	0,57±0,11
8-2	0,52±0,10	0,56±0,10	0,51±0,08	0,54±0,08	0,48±0,10	0,51±0,11	0,56±0,12	0,55±0,15	0,57±0,14	0,57±0,09
10-2	0,51±0,10	0,52±0,07	0,52±0,05	0,49±0,08	0,48±0,08	0,51±0,1	0,51±0,10	0,52±0,09	0,54±0,14	0,54±0,12
12-2	0,57±0,0	0,55±0,11	0,56±0,16	0,54±0,07	0,54±0,17	0,57±0,10	0,55±0,11	0,52±0,11	0,54±0,09	0,55±0,10
14-2	0,50±0,08	0,53±0,08	0,55±0,11	0,54±0,11	0,50±0,09	0,50±0,07	0,53±0,11	0,51±0,14	0,52±0,11	0,49±0,06
16-2	0,46±0,13	0,50±0,08	0,48±0,05	0,46±0,08	0,51±0,08	0,51±0,09	0,51±0,07	0,49±0,06	0,50±0,07	0,49±0,09

Fonte: Autor

Tabela 9 – Valores para o Momento de Zernike variando o número de repetições.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
4-2	**0,64±0,11**	**0,71±0,17**	**0,71±0,14**	**0,70±0,14**	**0,60±0,09**	**0,63±0,08**	**0,70±0,16**	**0,67±0,09**	**0,71±0,15**	**0,71±0,12**
4-4	0,55±0,13	0,57±0,10	0,56±0,12	0,59±0,10	0,54±0,11	0,57±0,12	0,56±0,15	0,54±0,16	0,59±0,10	0,60±0,09

Fonte: Autor

4.1.4 Utilizando a característica 1 - Histograma de Orientação de Borda combinado com a característica 2 - Centróide Hierárquico

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 10. Valor máximo em **0,96±0,03** para SVM Linear.

Tabela 10 – Utilização de Histograma de Orientação de Borda mais Centróide Hierárquico.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,88±0,09	**0,96±0,03**	0,91±0,07	0,97±0,03	0,86±0,05	0,90±0,09	0,97±0,04	0,94±0,03	0,90±0,05	0,86±0,08

Acur. = Acurácia Fonte: Autor

4.1.5 Utilizando a característica 1 - Histograma de Orientação de Borda combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 11. Valor máximo em **0,97±0,03** para Processo Gaussiano.

Tabela 11 – Utilização de Histograma de Orientação de Borda mais Momento de Zernike

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Árv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,87±0,03	0,94±0,06	0,90±0,07	**0,97±0,03**	0,85±0,07	0,86±0,04	0,94±0,06	0,90±0,05	0,85±0,12	0,87±0,07

Fonte: Autor

4.1.6 Utilizando a característica 2 - Centróide Hierárquico combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 12. Valor máximo em **0,87±0,07** para Floresta Aleatória.

Tabela 12 – Utilização de Centróide Hierárquico mais Momento de Zernike.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,83±0,04	0,82±0,05	0,80±0,03	0,85±0,04	0,81±0,06	**0,87±0,07**	0,83±0,04	0,84±0,06	0,83±0,04	0,82±0,07

Fonte: Autor

4.1.7 Utilizando todas as características agrupadas: Histograma de Orientação de Borda + Centróide Hierárquico + Momento de Zernike

Quando combinamos todas as característica, atingimos o patamar de acurácia, **0,97±0,02** para SVM Linear e Rede Neural, tabela 13.

Tabela 13 – Utilização de Histograma de Orientação de Borda, Centróide Hierárquico e Momento de Zernike

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,89±0,10	**0,97±0,02**	0,92±0,07	0,97±0,03	0,88±0,11	0,91±0,08	**0,97±0,02**	0,94±0,04	0,91±0,05	0,82±0,15

Fonte: Autor

4.1.8 Comparativo da utilização das características isoladas e em conjunto

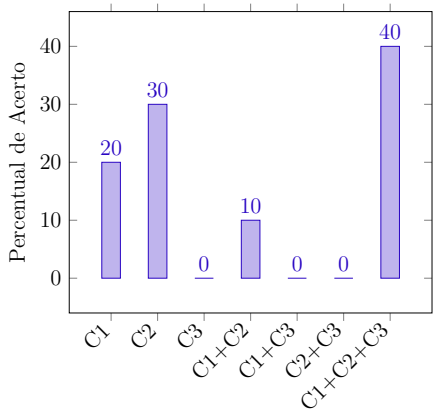
A tabela 14 mostra um comparativo na variação da acurácia em função da combinação de características. A figura 24 mostra que mesmo decaindo o valor para as três característica combinadas, ainda se mantém alto, precisando, acredita-se, num ajuste maior de parâmetros.

Tabela 14 – Características individuais e em conjunto para o Banco de Dados 1

Caract.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
C1	**0,93±0,09**	0,93±0,03	0,91±0,05	0,93±0,03	0,87±0,03	0,91±0,10	0,95±0,04	0,92±0,05	0,88±0,09	**0,92±0,06**
C2	0,90±0,05	0,89±0,04	0,88±0,04	0,92±0,05	**0,89±0,05**	**0,91±0,04**	0,89±0,05	0,92±0,04	**0,92±0,04**	0,86±0,04
C3	0,66±0,10	0,71±0,17	0,71±0,14	0,70±0,14	0,60±0,09	0,63±0,08	0,70±0,16	0,67±0,09	0,71±0,15	0,71±0,12
C1+C2	0,88±0,09	0,96±0,03	0,91±0,07	0,97±0,03	0,86±0,05	0,90±0,09	0,97±0,04	**0,94±0,03**	0,90±0,05	0,86±0,08
C1+C3	0,87±0,03	0,94±0,06	0,90±0,07	0,97±0,03	0,85±0,07	0,86±0,04	0,94±0,06	0,90±0,05	0,85±0,12	0,87±0,07
C2+C3	0,83±0,04	0,82±0,05	0,80±0,03	0,85±0,04	0,81±0,06	0,87±0,07	0,83±0,04	0,84±0,06	0,83±0,04	0,82±0,07
C1+C2+C3	0,89±0,10	**0,97±0,02**	**0,92±0,07**	**0,97±0,03**	0,88±0,11	0,91±0,08	**0,97±0,02**	0,94±0,04	0,91±0,05	0,82±0,15
{Maior Acurácia}	**0,93±0,09**	**0,97±0,02**	**0,92±0,07**	**0,97±0,03**	**0,89±0,05**	**0,91±0,04**	**0,97±0,02**	**0,94±0,03**	**0,92±0,04**	**0,92±0,06**

Fonte: Autor

Figura 24 – Valores de Máxima Acurácia comparado em percentuais para a combinação das características



4.2 Resultados Obtidos para o Banco de Dados 2 + Scikit-learn

4.2.1 Utilizando a característica 1 - Histograma de Orientação de Borda

Os parâmetros modificados aqui para uma melhoria na acurácia dos classificadores foram a quantidade de sub-imagens, variando de 1 até 100, e o tipo do filtro utilizado na função de cálculo do histograma (Canny, LoG, Prewitt, Roberts e Sobel). As tabelas mostraram a variação da acurácia em função do quantidade de sub-imagens (melhores classificadores) e tipo de filtro (para uma mesma quantidade de sub-imagens). Os classificadores escolhidos para comparação foram aqueles que tiveram uma melhor acurácia em toda a faixa de variação de sub-imagens (1-100). E o melhor classificador geral foi utilizado para um comparativo entre os tipos de filtros. Apesar de não ser necessário para o cálculo do histograma, as imagens foram normalizadas para uma dimensão de 200 x 200 pixels, igualando-as às utilizadas no cálculo das outras características. As tabelas 15, 16, 17, 18 e 19 mostram esses resultados. O melhor valor obtido foi com o filtro LoG (Laplaciano da Gaussiana) de 0,94±0,05 no classificador Processo Gaussiano, e utilizaremos esse filtro para mostrar a variação em função da quantidade de sub-imagens (figura 25).

4.2.1.1 Filtro Canny

A utilização do Filtro Canny (tabela 15) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0.92±0,05**, utilizando-se o classificadores SVM Linear e Processo Gaussiano e 45 valores de orientação.

Tabela 15 – Variação da Acurácia do Classificador em função da quantidade de sub-imagens utilizadas. Os valores exibidos na primeira coluna, formato S:C, representam a quantidade de cortes horizontais e verticais feitos na figura (S) e a quantidade de características geradas pelas sub-imagens (C)

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,89±0,06**	0,87±0,06	0,86±0,07	0,88±0,07	**0,83±0,08**	**0,88±0,06**	0,88±0,06	0,86±0,06	**0,81±0,06**	0,82±0,05
2-20	0,86±0,06	0,90±0,05	**0,87±0,06**	0,91±0,04	0,82±0,06	0,85±0,10	0,88±0,05	0,88±0,04	**0,81±0,06**	**0,84±0,05**
3-45	0,83±0,07	**0,92±0,05**	0,84±0,09	**0,92±0,05**	0,78±0,07	0,86±0,06	0,90±0,04	**0,88±0,03**	0,80±0,06	0,82±0,06
4-80	0,81±0,05	0,90±0,06	0,83±0,09	0,91±0,06	0,75±0,06	0,84±0,04	0,89±0,06	0,86±0,07	0,79±0,08	0,74±0,07
5-125	0,80±0,05	0,91±0,06	0,83±0,12	**0,92±0,05**	0,72±0,07	0,82±0,07	**0,92±0,06**	0,87±0,05	0,78±0,08	0,59±0,13
6-180	0,77±0,09	0,90±0,05	0,80±0,16	0,90±0,05	0,74±0,08	0,83±0,07	0,89±0,06	0,86±0,08	0,77±0,08	0,50±0,05
7-245	0,74±0,10	0,89±0,06	0,76±0,18	0,91±0,04	0,69±0,08	0,83±0,07	0,90±0,04	0,86±0,09	0,78±0,06	0,53±0,09
8-320	0,72±0,09	0,89±0,06	0,74±0,20	0,91±0,06	0,68±0,08	0,82±0,06	0,89±0,05	0,88±0,04	0,76±0,06	0,50±0,08
9-405	0,67±0,10	0,86±0,07	0,67±0,23	0,90±0,05	0,70±0,06	0,81±0,06	0,87±0,09	0,86±0,07	0,68±0,13	0,53±0,08
10-500	0,67±0,10	0,85±0,07	0,64±0,26	0,60±0,40	0,65±0,07	0,79±0,07	0,85±0,07	0,85±0,05	0,65±0,17	0,50±0,07

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão
Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

4.2.1.2 Filtro LoG - Laplacian of Gaussian

A utilização do Filtro LoG (tabela 16) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,94±0,05**, utilizando-se o classificador Processo Gaussiano com 125 valores de orientação.

Tabela 16 – Mesmo que a tabela 15, só que utilizando Filtro LoG

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,92±0,06**	0,90±0,05	0,88±0,07	0,93±0,05	**0,89±0,03**	**0,92±0,05**	0,90±0,06	0,90±0,04	**0,88±0,06**	**0,87±0,07**
2-20	0,87±0,06	0,92±0,05	**0,89±0,07**	0,93±0,04	0,84±0,10	0,89±0,03	0,91±0,06	**0,91±0,04**	0,87±0,08	0,86±0,09
3-45	0,83±0,06	0,91±0,05	0,86±0,08	0,92±0,04	0,80±0,06	0,87±0,07	0,91±0,06	0,91±0,05	0,84±0,06	0,82±0,07
4-80	0,80±0,05	0,91±0,04	0,86±0,09	0,92±0,04	0,80±0,03	0,88±0,05	0,92±0,05	0,90±0,06	0,81±0,07	0,72±0,11
5-125	0,80±0,04	**0,94±0,06**	0,84±0,11	**0,94±0,05**	0,78±0,07	0,86±0,08	**0,93±0,07**	0,90±0,05	0,78±0,08	0,57±0,10
6-180	0,76±0,08	0,92±0,04	0,81±0,14	0,93±0,05	0,77±0,07	0,85±0,03	0,92±0,05	0,90±0,05	0,76±0,06	0,50±0,07
7-245	0,74±0,09	0,90±0,05	0,79±0,13	0,92±0,05	0,75±0,09	0,86±0,05	0,91±0,04	0,89±0,05	0,74±0,06	0,50±0,09
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão
Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

4.2.1.3 Filtro Prewitt

A utilização do Filtro Prewitt (figura 17) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,88±0,06**, utilizando-se o classificador Processo Gaussiano e 45 valores de orientação.

Tabela 17 – Mesmo que na tabela 15, só que utilizando Filtro Prewitt

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,84±0,06**	0,85±0,05	**0,82±0,07**	0,86±0,06	**0,78±0,06**	**0,81±0,06**	0,85±0,05	0,79±0,07	**0,79±0,06**	**0,83±0,05**
2-20	0,78±0,07	0,85±0,05	**0,82±0,07**	0,87±0,06	0,76±0,09	0,79±0,07	**0,86±0,04**	0,80±0,03	0,78±0,05	0,80±0,06
3-45	0,75±0,10	**0,87±0,06**	0,80±0,08	**0,88±0,06**	0,73±0,07	0,79±0,06	0,86±0,07	0,80±0,06	0,76±0,06	0,77±0,06
4-80	0,71±0,08	0,83±0,05	0,78±0,07	0,84±0,05	0,69±0,08	0,76±0,07	0,84±0,05	0,80±0,06	0,75±0,06	0,72±0,06
5-125	0,70±0,07	0,83±0,06	0,76±0,10	0,85±0,07	0,69±0,06	0,77±0,08	0,83±0,06	**0,83±0,07**	0,74±0,07	0,59±0,16
6-180	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-245	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão
Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

4.2.1.4 Filtro Roberts

A utilização do Filtro Roberts (figura 18) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,84±0,04**, utilizando-se o classificador Processo Gaussiano e 45 valores de orientação.

Tabela 18 – Mesmo que na tabela 15, só que utilizando Filtro Roberts

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,74±0,06**	0,76±0,04	**0,72±0,06**	0,82±0,06	**0,75±0,06**	**0,76±0,06**	0,74±0,06	**0,77±0,06**	**0,72±0,07**	**0,75±0,05**
2-20	0,70±0,06	0,76±0,04	0,71±0,07	0,81±0,04	0,71±0,08	0,72±0,08	0,75±0,04	0,74±0,02	0,71±0,07	0,74±0,04
3-45	0,68±0,07	**0,81±0,05**	0,70±0,08	**0,84±0,04**	0,66±0,06	0,73±0,05	**0,79±0,07**	0,74±0,06	0,69±0,05	0,73±0,05
4-80	0,67±0,10	0,75±0,05	0,67±0,07	0,78±0,06	0,66±0,12	0,71±0,08	0,75±0,03	0,75±0,06	0,68±0,06	0,67±0,09
5-125	0,64±0,07	0,77±0,05	0,67±0,09	0,78±0,07	0,66±0,07	0,70±0,07	0,77±0,05	0,74±0,06	0,68±0,05	0,57±0,12
6-180	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-245	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão
Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

4.2.1.5 Filtro Sobel

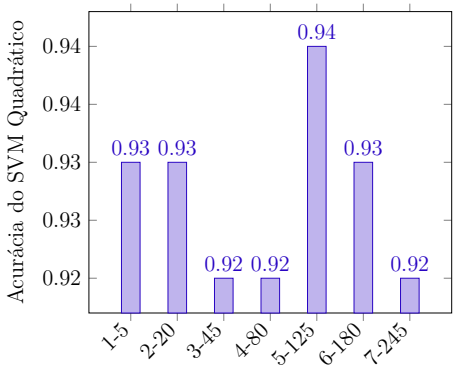
A utilização do Filtro Sobel (tabela 19) para geração da imagem em relevo para a determinação dos valores para o Histograma de borda, tem um máximo em **0,88±0,04**, utilizando-se o classificador Processo Gaussiano e 20 valores de orientação.

Tabela 19 – Mesmo que tabela 15, só que utilizando Filtro Sobel

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,84±0,05**	0,85±0,05	0,82±0,07	0,87±0,06	**0,80±0,05**	**0,81±0,05**	0,85±0,05	0,81±0,05	**0,80±0,06**	**0,83±0,05**
2-20	0,78±0,07	0,86±0,05	**0,83±0,07**	**0,88±0,04**	0,76±0,09	0,78±0,06	0,85±0,06	0,80±0,05	0,78±0,05	0,80±0,05
3-45	0,75±0,08	**0,87±0,07**	0,79±0,08	0,88±0,07	0,73±0,06	0,78±0,08	**0,86±0,06**	**0,82±0,05**	0,76±0,06	0,77±0,07
4-80	0,71±0,09	0,83±0,06	0,77±0,07	0,85±0,06	0,71±0,04	0,74±0,08	0,83±0,04	0,79±0,08	0,75±0,06	0,71±0,07
5-125	0,70±0,07	0,84±0,07	0,76±0,10	0,86±0,06	0,66±0,10	0,75±0,08	0,84±0,05	0,81±0,06	0,74±0,08	0,60±0,15
6-180	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7-245	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

KNN=K Nearest Neighbours SVM L.=SVM Linear S.RBF=SVM RBF P.Gaus.=Processo Gaussiano Arv.Dec.=Árvore de Decisão
Flo. Ale.=Floresta Aleatória R. N.=Rede Neural Adab.=Adaboost N. Bayes=Naive Bayes QDA=Discriminante Quadrático

Figura 25 – Filtro Processo Gaussiano X Classificador X Sub-Imagens



4.2.2 Utilizando a característica 2 - Centróide Hierárquico

Variaremos aqui a quantidade de sub-imagens utilizadas para o cálculo das centróides (tabela 20). O valor máximo obtido foi de **0,92±0,04** para o classificador *Naive Bayes*.

Tabela 20 – Centróide Hierárquico. A primeira coluna representa a quantidade de sub-imagens e a quantidade de características geradas por ela.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-4	0,69±0,06	0,49±0,09	0,49±0,09	0,73±0,07	0,65±0,07	0,66±0,08	0,60±0,12	0,70±0,05	0,70±0,07	0,73±0,06
3-12	0,90±0,06	0,89±0,05	0,88±0,05	0,91±0,05	**0,89±0,05**	**0,91±0,05**	**0,89±0,05**	**0,92±0,04**	0,88±0,05	**0,86±0,04**
4-28	0,89±0,04	0,89±0,06	0,88±0,05	**0,92±0,05**	**0,89±0,05**	0,90±0,06	**0,89±0,05**	0,91±0,05	0,90±0,05	0,83±0,04
5-60	0,90±0,06	**0,89±0,04**	0,87±0,05	0,91±0,04	0,85±0,06	0,90±0,05	**0,89±0,05**	0,90±0,02	**0,92±0,04**	0,77±0,06
6-124	**0,90±0,05**	0,88±0,05	**0,88±0,04**	0,91±0,04	0,84±0,05	0,89±0,05	0,89±0,06	0,91±0,05	0,91±0,03	0,60±0,17
7-252	0,89±0,05	0,88±0,05	0,87±0,05	0,91±0,04	0,85±0,07	0,89±0,06	**0,89±0,05**	0,89±0,05	0,91±0,04	0,50±0,08
8-508	0,89±0,05	0,87±0,04	0,87±0,04	0,91±0,04	0,84±0,05	0,89±0,05	0,89±0,06	0,89±0,05	0,91±0,04	0,50±0,11
9-1020	0,89±0,05	0,86±0,04	0,87±0,04	0,91±0,04	0,84±0,06	0,89±0,07	0,88±0,06	0,90±0,04	0,91±0,04	0,50±0,09
10-2044	0,89±0,05	0,86±0,05	0,87±0,06	0,91±0,03	0,84±0,04	0,90±0,03	0,88±0,05	0,90±0,05	0,91±0,04	0,51±0,09

Fonte: Autor

4.2.3 Utilizando a característica 3 - Momentos de Zernike

Para esta característica analisaremos duas situações:

1. variando a ordem do polinômio (2-16, ordem par - tabela 21)
2. a quantidade de repetições do Momento de Zernike (2-4, ordem par - tabela 22)

Os valores máximos obtidos foram **0,78±0,05** e **0,77±0,07**, ambos para Processo Gaussiano.

Tabela 21 – Momento de Zernike variando a ordem.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-2	**0,76±0,07**	**0,75±0,05**	**0,75±0,06**	**0,78±0,05**	**0,69±0,08**	**0,73±0,05**	**0,73±0,07**	**0,73±0,04**	0,76±0,03	0,74±0,07
4-2	0,76±0,08	0,75±0,08	**0,75±0,06**	0,77±0,07	0,68±0,09	0,72±0,10	0,73±0,08	0,73±0,08	**0,77±0,04**	**0,76±0,04**
6-2	0,60±0,05	0,62±0,06	0,59±0,08	0,61±0,09	0,55±0,08	0,56±0,07	0,61±0,08	0,59±0,06	0,62±0,08	0,61±0,07
8-2	0,52±0,09	0,59±0,07	0,57±0,09	0,58±0,07	0,50±0,07	0,51±0,07	0,57±0,08	0,55±0,09	0,59±0,04	0,57±0,05
10-2	0,52±0,07	0,53±0,09	0,50±0,09	0,50±0,09	0,54±0,08	0,53±0,07	0,53±0,08	0,53±0,05	0,52±0,10	0,52±0,07
12-2	0,53±0,07	0,52±0,08	0,47±0,04	0,53±0,08	0,53±0,07	0,55±0,06	0,50±0,09	0,56±0,06	0,50±0,09	0,52±0,09
14-2	0,51±0,07	0,50±0,12	0,48±0,06	0,48±0,06	0,51±0,08	0,51±0,07	0,50±0,12	0,51±0,10	0,51±0,10	0,51±0,11
16-2	0,49±0,09	0,53±0,10	0,50±0,09	0,49±0,09	0,50±0,07	0,49±0,09	0,53±0,10	0,50±0,10	0,52±0,10	0,52±0,09

Fonte: Autor

Tabela 22 – Momento de Zernike variando a quantidade de repetições.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
4-2	**0,76±0,08**	**0,75±0,08**	**0,75±0,06**	**0,77±0,07**	**0,68±0,09**	**0,71±0,07**	**0,74±0,06**	**0,73±0,08**	**0,77±0,04**	**0,76±0,04**
4-4	0,58±0,11	0,61±0,05	0,60±0,05	0,61±0,05	0,56±0,11	0,57±0,06	0,61±0,04	0,57±0,06	0,61±0,06	0,61±0,04

Fonte: Autor

4.2.4 Utilizando a característica 1 - Histograma de Orientação de Borda combinada com a característica 2 - Centróide Hierárquico

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 23. Valor máximo em **0,96±0,03** para SVM Linear e Processo Gaussiano.

Tabela 23 – Utilização de Histograma de Orientação de Borda mais Centróide Hierárquico.

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,88±0,05	**0,96±0,03**	0,93±0,05	**0,96±0,03**	0,90±0,06	0,94±0,04	0,95±0,03	0,95±0,04	0,90±0,05	0,85±0,08

Acur. = Acurácia Fonte: Autor

4.2.5 Utilizando a característica 1 - Histograma de Orientação de Borda combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 24. Valor máximo em **0,96±0,03** para SVM Linear.

Tabela 24 – Utilização de Histograma de Orientação de Borda mais Momento de Zernike

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,88±0,07	**0,96±0,03**	0,92±0,06	0,96±0,04	0,83±0,08	0,90±0,05	0,95±0,04	0,95±0,05	0,88±0,06	0,85±0,07

Acur. = Acurácia Fonte: Autor

4.2.6 Utilizando a característica 2 - Centróide Hierárquico combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 25. Valor máximo em **0,96±0,03** para SVM Linear.

Tabela 25 – Utilização de Centróide Hierárquico mais Momento de Zernike.

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,88±0,07	**0,96±0,03**	0,92±0,06	0,96±0,04	0,83±0,08	0,90±0,05	0,95±0,04	0,95±0,05	0,88±0,06	0,85±0,07

Acur. = Acurácia Fonte: Autor

4.2.7 Utilizando todas as características agrupadas: Histograma de Orientação de Borda + Centróide Hierárquico + Momento de Zernike

Quando combinamos todas as característica, atingimos o patamar de acurácia, **0,97±0,02** para SVM Linear e Processo Gaussiano, tabela 26.

Tabela 26 – Utilização de Histograma de Orientação de Borda, Centróide Hierárquico e Momento de Zernike

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,91±0,08	**0,97±0,02**	0,94±0,04	**0,97±0,02**	0,89±0,06	0,93±0,06	0,96±0,05	0,95±0,03	0,92±0,06	0,85±0,08

Acur. = Acurácia Fonte: Autor

4.2.8 Características individuais e em conjunto para o Banco de Dados 2

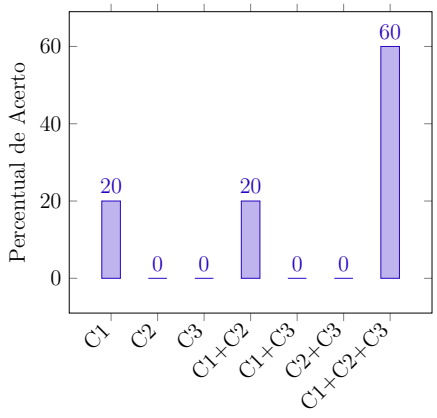
A tabela 27 faz um comparativo entre a utilização individual, combinação de duas e finalmente as três características agrupados. A figura 26 mostra mais uma vez que a utilização de várias características agrupadas levam a uma elevação na acurácia do reconhecimento das rasuras.

Tabela 27 – Características individuais e em conjunto para o Banco de Dados 2

Caract.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
C1	**0,92±0,06**	0,94±0,06	0,89±0,07	0,94±0,05	0,89±0,03	0,92±0,05	0,93±0,07	0,91±0,04	0,88±0,06	**0,87±0,07**
C2	0,90±0,05	0,89±0,04	0,88±0,04	0,92±0,05	0,89±0,05	0,91±0,05	0,89±0,05	0,92±0,04	0,92±0,07	0,86±0,04
C3	0,76±0,07	0,75±0,05	0,75±0,06	0,78±0,05	0,69±0,08	0,73±0,05	0,73±0,07	0,73±0,04	0,77±0,04	0,76±0,04
C1+C2	0,88±0,05	0,96±0,03	0,93±0,05	0,96±0,03	**0,90±0,06**	**0,94±0,04**	0,95±0,03	0,95±0,04	0,90±0,05	0,85±0,08
C1+C3	0,88±0,07	0,96±0,03	0,92±0,06	0,96±0,04	0,83±0,08	0,90±0,05	0,95±0,04	0,95±0,05	0,88±0,06	0,85±0,07
C2+C3	0,88±0,07	0,96±0,03	0,92±0,06	0,96±0,04	0,83±0,08	0,90±0,05	0,95±0,04	0,95±0,05	0,88±0,06	0,85±0,07
C1+C2+C3	0,91±0,08	**0,97±0,02**	**0,94±0,04**	**0,97±0,02**	0,89±0,06	0,93±0,06	**0,96±0,05**	**0,95±0,03**	**0,92±0,06**	0,85±0,08
Maior Acurácia	**0,92±0,06**	**0,97±0,02**	**0,94±0,04**	**0,97±0,02**	**0,90±0,06**	**0,94±0,04**	**0,96±0,05**	**0,95±0,03**	**0,92±0,06**	**0,87±0,07**

Fonte: Autor

Figura 26 – Valores de Máxima Acurácia comparado em percentuais para a combinação das características



4.3 Resultados Obtidos para o Banco de Dados 3 + Scikit-learn

4.3.1 Utilizando a característica 1 - Histograma de Orientação de Borda

Os parâmetros modificados aqui para uma melhoria na acurácia dos classificadores foram a quantidade de sub-imagens, variando de 1 até 100, e o tipo do filtro utilizado na função de cálculo do histograma (Canny, LoG, Prewitt, Roberts e Sobel). As tabelas mostraram a variação da acurácia em função do quantidade de sub-imagens (melhores classificadores) e tipo de filtro (para uma mesma quantidade de sub-imagens). Os classificadores escolhidos para comparação foram aqueles que tiveram uma melhor acurácia em toda a faixa de variação de sub-imagens (1-100). E o melhor classificador geral foi utilizado para um comparativo entre os tipos de filtros. Apesar de não ser necessário para o cálculo do histograma, as imagens foram normalizadas para uma dimensão de 200 x 200 pixels, igualando-as às utilizadas no cálculo das outras características. A tabela 28, mostra esses resultados. Utilizamos o filtro LoG e uma quantidade de amostras de 10000. O melhor valor obtido foi de **0,94±0,05** para o classificador Processo Gaussiano.

Tabela 28 – Valores de acurácia para conjunto de 10000 amostras e filtro LoG.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
1-5	**0,92±0,06**	0,90±0,05	0,88±0,07	0,93±0,05	**0,89±0,03**	**0,93±0,04**	0,90±0,06	0,90±0,04	**0,88±0,06**	**0,87±0,07**
2-20	0,87±0,06	0,92±0,05	**0,89±0,07**	0,93±0,04	0,84±0,10	0,90±0,05	0,91±0,06	**0,91±0,04**	0,87±0,08	0,86±0,09
3-45	0,83±0,06	0,91±0,05	0,86±0,08	0,92±0,04	0,80±0,06	0,89±0,06	0,91±0,07	0,91±0,05	0,84±0,06	0,82±0,07
4-80	0,80±0,05	0,91±0,04	0,86±0,09	0,92±0,04	0,80±0,03	0,89±0,04	0,92±0,05	0,90±0,06	0,81±0,07	0,72±0,11
5-125	0,80±0,04	0,94±0,06	0,84±0,11	**0,94±0,05**	0,78±0,07	0,87±0,05	**0,93±0,06**	0,90±0,05	0,78±0,08	0,57±0,10
6-180	0,76±0,08	**0,92±0,04**	0,81±0,14	0,93±0,05	0,77±0,07	0,87±0,06	0,92±0,06	0,90±0,05	0,76±0,06	0,50±0,07
7-245	0,74±0,09	0,90±0,05	0,79±0,13	0,92±0,05	0,75±0,09	0,84±0,06	0,91±0,05	0,89±0,05	0,74±0,06	0,50±0,09
8-320	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-405	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-500	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Fonte: Autor

4.3.2 Utilizando a característica 2 - Centróide Hierárquico

Variaremos aqui a quantidade de sub-imagens utilizadas para o cálculo das centróides (tabela 29). O valor máximo obtido foi de **0,95±0,01** para o classificador Discriminante Quadrático.

Tabela 29 – Centróide Hierárquico. A primeira coluna representa a quantidade de sub-imagens e a quantidade de características geradas por ela.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-4	0,44±0,01	0,31±0,02	0,26±0,03	N/A	0,36±0,02	0,41±0,02	0,23±0,01	0,29±0,03	0,23±0,02	0,42±0,01
3-12	0,84±0,01	0,72±0,02	0,76±0,02	N/A	0,72±0,01	0,81±0,02	0,72±0,02	0,47±0,20	0,72±0,01	0,83±0,01
4-28	0,89±0,01	0,84±0,01	0,83±0,01	N/A	0,77±0,01	0,87±0,01	0,81±0,01	0,45±0,19	0,74±0,01	0,92±0,01
5-60	**0,92±0,01**	0,93±0,01	**0,87±0,01**	N/A	0,82±0,02	0,90±0,01	0,89±0,01	0,49±0,14	**0,82±0,01**	**0,95±0,01**
6-124	**0,92±0,01**	**0,94±0,01**	**0,87±0,01**	N/A	**0,82±0,01**	**0,91±0,01**	**0,90±0,01**	**0,50±0,12**	0,80±0,01	**0,95±0,01**
7-252	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8-508	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9-1020	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10-2044	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

4.3.3 Utilizando a característica 3 - Momentos de Zernike

Para esta característica analisaremos duas situações:

1. variando a ordem do polinômio (2-16, ordem par - tabela 30)

2. a quantidade de repetições do Momento de Zernike (2-4, ordem par - tabela 31)

Os valores máximos obtidos foram **0,32±0,02** para SVM RBF e *Naive Bayes* e **0,31±0,01** para Rede Neural.

Tabela 30 – Momento de Zernike variando a ordem.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
2-2	0,26±0,02	0,28±0,01	0,29±0,02	N/A	0,20±0,02	0,22±0,02	0,28±0,02	0,26±0,01	0,29±0,02	0,29±0,02
4-2	0,28±0,02	0,30±0,01	0,31±0,02	N/A	0,21±0,01	0,24±0,01	0,30±0,01	0,26±0,03	0,31±0,02	0,30±0,02
6-2	**0,30±0,01**	**0,31±0,02**	**0,32±0,02**	N/A	**0,23±0,01**	**0,26±0,01**	**0,31±0,02**	**0,28±0,04**	**0,32±0,02**	**0,32±0,01**
8-2	0,24±0,02	0,26±0,01	0,26±0,01	N/A	0,17±0,01	0,20±0,01	0,25±0,01	0,24±0,02	0,27±0,01	0,27±0,01
10-2	0,19±0,01	0,23±0,01	0,23±0,01	N/A	0,14±0,01	0,16±0,01	0,21±0,01	0,21±0,01	0,23±0,01	0,23±0,01
12-2	0,15±0,01	0,19±0,01	0,19±0,01	N/A	0,13±0,01	0,14±0,01	0,18±0,01	0,19±0,01	0,20±0,01	0,20±0,01
14-2	0,15±0,01	0,19±0,01	0,19±0,02	N/A	0,12±0,01	0,13±0,01	0,18±0,02	0,18±0,02	0,19±0,01	0,19±0,02
16-2	0,15±0,01	0,19±0,01	0,19±0,01	N/A	0,13±0,01	0,13±0,01	0,18±0,02	0,18±0,01	0,20±0,01	0,19±0,01

Fonte: Autor

Tabela 31 – Momento de Zernike variando a quantidade de repetições.

S./C.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
4-2	**0,28±0,02**	**0,30±0,01**	**0,31±0,02**	N/A	**0,21±0,01**	**0,24±0,01**	**0,31±0,01**	0,26±0,03	**0,31±0,02**	**0,30±0,02**
4-4	0,26±0,01	0,28±0,01	0,29±0,01	N/A	0,20±0,01	0,23±0,01	0,28±0,00	**0,28±0,02**	0,29±0,01	0,29±0,01

Fonte: Autor

4.3.4 Utilizando a característica 1 - Histograma de Orientação de Borda combinada com a característica 2 - Centróide Hierárquico

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 32. Valor máximo em **0,31±0,01** para SVM Linear.

Tabela 32 – Utilização de Histograma de Orientação de Borda mais Centróide Hierárquico.

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,94±0,03	**0,96±0,02**	0,93±0,03	N/A	0,83±0,03	0,91±0,04	0,94±0,02	0,49±0,10	0,86±0,04	0,91±0,06

Acur. = Acurácia Fonte: Autor

4.3.5 Utilizando a característica 1 - Histograma de Orientação de Borda combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 33. Valor máximo em **0,94±0,03** para SVM Linear.

Tabela 33 – Utilização de Histograma de Orientação de Borda mais Momento de Zernike

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	0,91±0,03	**0,94±0,03**	0,92±0,04	N/A	0,78±0,03	0,89±0,04	0,92±0,02	0,56±0,12	0,82±0,05	0,83±0,06

Acur. = Acurácia Fonte: Autor

4.3.6 Utilizando a característica 2 - Centróide Hierárquico combinada com a característica 3 - Momento de Zernike

Quando começamos a agregar as características utilizadas para a geração das características das imagens, começamos a perceber um incremento na acurácia, tabela 34. Valor máximo em **0,83±0,06** para o classificador KNN.

Tabela 34 – Utilização de Centróide Hierárquico mais Momento de Zernike.

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	**0,83±0,06**	0,74±0,06	0,76±0,06	N/A	0,71±0,04	0,81±0,04	0,67±0,08	0,46±0,19	0,72±0,04	0,84±0,04

Acur. = Acurácia Fonte: Autor

4.3.7 Utilizando todas as características agrupadas: Histograma de Orientação de Borda + Centróide Hierárquico + Momento de Zernike

Quando combinamos todas as característica, atingimos o patamar de acurácia, **0,93±0,04** para KNN e SVM Linear, tabela 35.

Tabela 35 – Utilização de Histograma de Orientação de Borda, Centróide Hierárquico e Momento de Zernike

S./C	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
Acur.	**0,93±0,04**	**0,93±0,04**	0,92±0,05	N/A	0,81±0,04	0,90±0,04	0,90±0,03	0,62±0,13	0,86±0,06	0,92±0,04

Acur. = Acurácia Fonte: Autor

4.3.8 Características individuais e em conjunto para o Banco de Dados 3

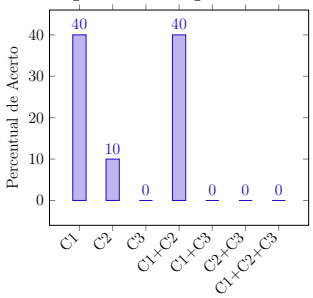
A tabela 36 faz um comparativo entre a utilização individual, combinação de duas e finalmente as três características agrupadas. A figura 27 mostra mais uma vez a eficiência da utilização da combinação de características.

Tabela 36 – Características individuais e em conjunto para o Banco de Dados 3

Caract.	KNN	SVM L.	S.RBF	P.Gaus.	Arv.Dec.	Flo.Ale.	R. N.	Adab.	N.Bayes	QDA
C1	0,92±0,06	0,92±0,04	0,89±0,07	N/A	**0,89±0,03**	**0,93±0,04**	0,90±0,06	**0,90±0,04**	**0,88±0,06**	0,87±0,07
C2	0,92±0,01	0,94±0,01	0,87±0,01	N/A	0,82±0,01	0,91±0,01	0,90±0,01	0,50±0,12	0,80±0,01	**0,95±0,01**
C3	0,30±0,01	0,31±0,02	0,32±0,02	N/A	0,23±0,01	0,26±0,01	0,31±0,02	0,28±0,02	0,32±0,02	0,32±0,01
C1+C2	**0,94±0,03**	**0,96±0,02**	**0,93±0,03**	N/A	0,83±0,03	0,91±0,04	**0,94±0,02**	0,49±0,10	0,86±0,04	0,91±0,06
C1+C3	0,91±0,03	0,94±0,03	0,92±0,04	N/A	0,78±0,043	0,89±0,04	0,92±0,02	0,56±0,12	0,82±0,05	0,83±0,06
C2+C3	0,83±0,06	0,74±0,06	0,76±0,06	N/A	0,71±0,04	0,81±0,04	0,67±0,08	0,46±0,19	0,72±0,04	0,84±0,04
C1+C2+C3	0,93±0,04	0,93±0,04	0,93±0,05	N/A	0,81±0,04	0,90±0,04	0,90±0,03	0,62±0,13	0,86±0,06	0,92±0,04
Maior Acurácia	**0,94±0,03**	**0,96±0,02**	**0,93±0,03**	N/A	**0,89±0,03**	**0,93±0,04**	**0,94±0,02**	**0,90±0,04**	**0,88±0,06**	**0,95±0,01**

Fonte: Autor

Figura 27 – Valores de Máxima Acurácia comparado em percentuais para a combinação das características



4.4 Considerações Finais

Consegue-se provar que a utilização de características combinadas, desde que bem escolhidas e os classificadores bem ajustados, podem melhorar a acurácia dos classificadores. Os resultados obtidos para o banco de dados 3 (caracteres), levam a crer que esse método não seria adequado para implementação de OCR (*Optical Character Recognitian* - Reconhecimento Ótico de Caracteres). Para esse propósito, estudos maiores deverão ser feitos. Estudos feitos com *Deep Learning* poderiam levar à resultados semelhantes ou até melhores, já que *Deep Learning* pressupõe a descoberta de um de conjunto de características ótimas para a classificação. A proposta foi tentar atingir uma boa relação entre a acurácia na classificação (identificação das rasuras) e o custo computacional.

5 Conclusões

Nesta dissertação apresentou-se uma proposta de reconhecimento de rasuras através da utilização de processos de aprendizado de máquina (aprendizado supervisionado) e a combinação de descritores de imagens diferentes (de direção, de forma, de intensidade). Foram feitas classificações com diversos algoritmos (SVM, KNN, Gaussian SVM, etc) e observado-se a evolução dos resultados em função da combinação das características geradas pelos descritores.

Os resultados obtidos são interessantes. Verificou-se que os descritores de imagens quando utilizados sozinhos atingem limites máximos de acurácia da ordem de 90%, mas combinados aos outros chegam a ultrapassar os 98% de acurácia. A utilização de descritores com características diferenciadas, gera características que combinadas, descrevem melhor a imagem. Descritores que isoladamente, contribuem apenas para uma acurácia entre 70-80% elevam essa acurácia para valores entre 96-98% quando combinados.

Como prolongamento imediato do trabalho aqui proposto, pode-se ser feito um estudo maior das rasuras, em grupos de crianças com problemas comprovados de dificuldade de aprendizagem, para determinação de padrões específicos. Esses padrões podem ser utilizados em sistemas de aprendizado de máquina como uma forma de determinação de transtornos de aprendizagem, tais como a dislexia, discalculia e outros, ainda no início do processo de aprendizagem.

Referências

ASIRI NORAH M.; ALHUMAIDI, N. A. N. [ieee 2015 second international conference on computer science, computer engineering, and social media (cscesm) - lodz, poland (2015.9.21-2015.9.23)] 2015 second international conference on computer science, computer engineering, and social media (cscesm) - combination of histogram of oriented gradients and hierarchical centroid for sketch-based image retrieval. In: . [S.l.: s.n.], 2015. ISBN 978-1-4799-1790-7. Citado na página 28.

BREIMAN JEROME FRIEDMAN, R. A. O. C. J. S. L. *Classification and regression trees*. [S.l.]: CRC, 1984. (The Wadsworth statistics / probability series). ISBN 0-412-04841-8. Citado na página 47.

CALIL, E. *Escutar o invisivel: escritura & poesia na sala de aula*. Unesp, 2008. (Coleção arte e educação). ISBN 9788571398016. Disponível em: <https://books.google.com.br/books?id=nu2VPgAACAAJ>. Citado 4 vezes nas páginas 19, 22, 23 e 24.

CAPRISTANO, C. *Mudanças na trajetória da criança em direção à palavra escrita*. Tese (Doutorado) — Universidade Estadual de Campinas, Instituto de Estudos da Linguagem, Pós-Graduação em Lingüística Aplicada, Campinas, 2007. Citado na página 19.

CAPRISTANO, C.; CHACON, L. Relações metafóricas e metonímicas: notas sobre a "aquisição" da noção de palavra. *O (In) esperado de Jakobson. Campinas: Mercado de Letras, prelo*, 2014. Citado na página 19.

CôRREA, M. L. G. *O Modo Heterogêneo de Constituição da Escrita*. [S.l.]: Martins Fontes, 2004. ISBN 9788533620094. Citado na página 19.

FELIPETO, C. *Rasura e Equívoco - no processo de escritura em sala de aula*. Eduel, 2008. ISBN 9788572164863. Disponível em: <https://books.google.com.br/books?id=Sx6Cw8_tJ7QC>. Citado 2 vezes nas páginas 19 e 22.

HARRINGTON, P. *Machine Learning in Action*. [S.l.]: Manning Publications, 2012. ISBN 1617290181,9781617290183. Citado 3 vezes nas páginas 35, 36 e 37.

HAYKIN, S. O. *Neural Networks and Learning Machines*. 3. ed. [S.l.]: Prentice Hall, 2008. (3rd Edition). ISBN 0131471392,9780131471399. Citado na página 34.

LECUN Y.; BOTTOU, L. B. Y. H. P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, 1998. Citado na página 43.

MERCER, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London Series A Containing Papers of a Mathematical or Physical Character (18*, The Royal Society, v. 209, 1909. Citado na página 32.

MITCHELL, T. M. *Machine Learning*. 1. ed. [S.l.]: McGraw-Hill, 1997. (McGraw-Hill series in computer science). ISBN 9780070428072,0070428077. Citado 2 vezes nas páginas 31 e 32.

MüLLER, S. G. A. C. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. 1. ed. [S.l.]: O'Reilly Media, 2016. ISBN 1449369413,9781449369415. Citado na página 30.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 31 e 38.

PEDRINI, H.; SCHWARTZ, W. *Análise de imagens digitais: princípios, algoritmos e aplicações*. THOMSON PIONEIRA, 2008. ISBN 9788522105953. Disponível em: <https://books.google.com.br/books?id=13KAPgAACAAJ>. Citado na página 26.

SAKI, F. et al. Fast opposite weight learning rules with application in breast cancer diagnosis. *Computers in Biology and Medicine*, Elsevier, v. 43, n. 1, p. 32–41, 2013. ISSN 00104825. Disponível em: <http://dx.doi.org/10.1016/j.compbiomed.2012.10.006>. Citado na página 40.

STAMP, M. *Introduction to Machine Learning with Applications in Information Security*. [S.l.]: CRC, 2017. ISBN 978-1-138-62678-2. Citado na página 32.

SUTHAHARAN, S. *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. 1. ed. [S.l.]: Springer US, 2016. (Integrated Series in Information Systems 36). ISBN 978-1-4899-7640-6,978-1-4899-7641-3. Citado 2 vezes nas páginas 29 e 36.

TAHMASBI, A.; SAKI, F.; SHOKOUHI, S. B. Classification of benign and malignant masses based on Zernike moments. *Computers in Biology and Medicine*, Elsevier, v. 41, n. 8, p. 726–735, 2011. ISSN 00104825. Disponível em: <http://dx.doi.org/10.1016/j.compbiomed.2011.06.009>. Citado na página 40.

WILLEMART, P. *Universo da criação literária: crítica genética, crítica pós-moderna?* EDUSP, 1993. (Criação & crítica). ISBN 9788531401268. Disponível em: <https://books.google.com.br/books?id=-u1l9Ynx9g8C>. Citado na página 19.

ZHENJIANG, M. Zernike moment-based image shape analysis and its application. *Pattern Recognition Letters*, Elsevier Science, v. 21, 2000. Citado na página 26.

Apêndices

APÊNDICE A – Códigos Fontes dos Programas

A.1 Histograma de Orientação de Borda

```
% Função para calcular o histograma de orientação de borda
% Baseado em Edge orientation histograms in global and local features
% (Octave/Matlab) por Dr. Roberto Ulloa em
% http://robertour.com/2012/01/26/edge-orientation-histograms-in-global-and-local-features/
% parâmetros
% im - a imagem
% r - o número divisões verticais e horizontais
% funcao - função de detecção de borda ('canny', 'log', 'prewitt',
% 'roberts', 'sobel'

function [data] = edgeOrientationHistogram(im, r,funcao)

% define os filtros para os 5 tipos de borda
f2 = zeros(3,3,5);
f2(:,:,1) = [1 2 1;0 0 0;-1 -2 -1]; % vertical
f2(:,:,2) = [-1 0 1;-2 0 2;-1 0 1]; % horizontal
f2(:,:,3) = [2 2 -1;2 -1 -1; -1 -1 -1]; % diagonals
f2(:,:,4) = [-1 2 2; -1 -1 2; -1 -1 -1];
f2(:,:,5) = [-1 0 1;0 0 0;1 0 -1]; %  non-directional

% o tamanho da imagem
ys = size(im,1);
xs = size(im,2);

% Constrói uma nova matriz das mesmas dimensões da imagem
% e 5 dimensões para salvar os gradientes
im2 = zeros(ys,xs,5);

% interage sobre as possíveis direções
for i = 1:5
    % aplica a máscara de sobel
    im2(:,:,i) = filter2(f2(:,:,i), im);
end

% calcula o máximo gradiente de sobel
[mmax, maxp] = max(im2,[],3);
% salva somente o índice (tipo) da direção
% e ignora o valor do gradiente
im2 = maxp;

% detecta as bordas usando os parâmetros padrões do Matlab
ime = edge(im, funcao);

% multiplica as orientações detectadas
% pelas máscaras de Sobel
im2 = im2.*ime;
```

```
% produz uma estrutura para salvar todos as faixas do histograma
% de cada região
eoh = zeros(r,r,6);
% for each region
for j = 1:r
    for i = 1:r
        % extrai a subimagem
        clip = im2(round((j-1)*ys/r+1):round(j*ys/r),round((i-1)*xs/r+1):round(i*xs/r));
        % calcula o histograma para a região
        eoh(j,i,:) = (hist(makelinear(clip), 0:5)*100)/numel(clip);
    end
end

% retira os zeros
eoh = eoh(:,:,2:6);

% representa todos os histogramas em um vetor
data = zeros(1,numel(eoh));
data(:) = eoh(:);
end
```

A.2 Hierarchical Centroid

```
% Descriptor for simple shapes (e.g letters).
% Computes the descriptor twice in order to describe
% horizontally and vertically with the same accuracy.
%
% Input: image (2-D, binary or gray)
%        depth of recursion (default 7),
%        plotFlag(if on - plot Illustration)
% Output:
%        vec - descriptor: Division locations, values around zero. Of length 2*(2^depth - 2)
%        (levels: the depth of the division locations.)
%
% Shahar Armon. 3/1/2012

function  [vec levels] = hierarchicalCentroid(im, depth, plotFlag)
    if nargin < 2
        depth = 7;
        plotFlag = 0;
    elseif nargin < 3
        plotFlag = 0;
    end

    im(im < 0) = 0;

    [vec1 levels1] = hierarchicalCentroid1(im  ,depth, plotFlag);
    [vec2 levels2] = hierarchicalCentroid1(im' ,depth, 0);  % transpose iamge

    vec = [vec1, vec2];
    levels = [levels1,-levels2];  % Minus is only to mark the transpose.
end


% The descriptor. Includes Normalization.
function  [vec levels] = hierarchicalCentroid1(im, d, plotFlag)
```

```
    p = hierarchicalCentroidRec(im, 1, d);

    if plotFlag  % Illustration
        showLines(im, p);
    end

    meany = ([1:size(im,1)]*sum(im,2))/sum(sum(im));
    indVer = (mod(p(:,2),2) == 1);
    % Normalization for location:
    p(indVer,1) = p(indVer,1) - p(1,1);
    p(~indVer,1) = p(~indVer,1) - meany;

    % Normalization for size (keeping aspect ratio):
    p(:,1) = p(:,1)/size(im,1);
    % Normalization for size:
    % p(indVer,1) = p(indVer,1)/size(im,2);
    % p(~indVer,1) = p(~indVer,1)/size(im,1);

    % Illustration of the lines after normalization without the image
    %     showLines([], p);

    vec = p(2:end,1)';
    levels = p(2:end,2)';
end



function p = hierarchicalCentroidRec(im, depth, maxDepth)
    if depth > maxDepth
        p = [];
    else
        area = sum(sum(im));
        [rows,cols] = size(im);
        % compute the centroid-x
        if cols == 1
            centroid = 0.5;
        elseif area == 0
            centroid = cols/2;
        elseif rows == 1
            centroid = (im*[1:cols]')/area;
        else
            centroid = sum(im)*[1:cols]'/area;
        end

        leftIm = im(:,1:floor(centroid));
        rightIm = im(:,ceil(centroid):end);


        pLeft  = hierarchicalCentroidRec(leftIm'  , depth+1, maxDepth);
        pRight = hierarchicalCentroidRec(rightIm' , depth+1, maxDepth);

        %  Updates the distances so they will be in relation to the complete image
        if size(pRight,1) > 1
            ind = find((1-mod(depth,2))-mod(pRight(2:end,2),2)) + 1;
            pRight(ind,1) = pRight(ind,1) + ceil(centroid) - 1;
            pRight(1,1) = pRight(1,1) - 1;
        end

        p = [centroid, depth; pLeft; pRight];
    end
```

```
end



function showLines(im, p)
    if max(p(:,1))<3
        hold on;
        axis([-1 1 -1 1])
        axis ij;
        temp = mod(p(:,2),2);
        indVer = find(temp);
        indHor = find(~temp);
        showLinesRec(min(p(indVer,1)),max(p(indVer,1)),min(p(indHor,1)),max(p(indHor,1)), p);
    else
        imshow(im);
        hold on;
        showLinesRec(0,size(im,2),0,size(im,1), p);
    end
    hold off
end

function p = showLinesRec(x1,x2,y1,y2, p)
    p1 = p(1,:);
    w = 1+max(p(:,2))-p1(1,2);
    t = p1(1);
    p = p(2:end,:);
    if mod(p1(1,2),2)
        plot([t t],[y1 y2],'-m','LineWidth',w);
    else
        plot([x1 x2],[t t],'-r','LineWidth',w);
    end
    if (p1(1,2) < max(p(:,2)))
        if mod(p1(1,2),2)
            p = showLinesRec(x1,t,y1,y2, p);
            p = showLinesRec(t,x2,y1,y2, p);
        else
            p = showLinesRec(x1,x2,y1,t, p);
            p = showLinesRec(x1,x2,t,y2, p);
        end
    end

end
```

A.3 Zernike Moment

```
# ---------------------------------------------------------------------
# Copyright C 2015 Gefu Tang
# tanggefu@gmail.com
#
# License Agreement: To acknowledge the use of the code please cite the
#                    following papers:
#
# [1] A. Tahmasbi, F. Saki, S. B. Shokouhi,
#     Classification of Benign and Malignant Masses Based on Zernike Moments,
#     Comput. Biol. Med., vol. 41, no. 8, pp. 726-735, 2011.
#
```

```
# [2] F. Saki, A. Tahmasbi, H. Soltanian-Zadeh, S. B. Shokouhi,
#     Fast opposite weight learning rules with application in breast cancer
#     diagnosis, Comput. Biol. Med., vol. 43, no. 1, pp. 32-41, 2013.
# ------------------------------------------------------------------------
import numpy as np
from math import factorial


# ------------------------------------------------------------------------
# Function to compute Zernike Polynomials:
#
# rad = radialpoly(r,n,m)
# where
#   r = radius
#   n = the order of Zernike polynomial
#   m = the repetition of Zernike moment
# ------------------------------------------------------------------------
def radialpoly(r, n, m):
    rad = np.zeros(r.shape, r.dtype)
    P = (n - abs(m)) / 2
    Q = (n + abs(m)) / 2
    for s in xrange(P + 1):
        c = (-1) ** s * factorial(n - s)
        c /= factorial(s) * factorial(Q - s) * factorial(P - s)
        rad += c * r ** (n - 2 * s)
    return rad


# ------------------------------------------------------------------------
# Function to find the Zernike moments for an N x N binary ROI
#
# Z, A, Phi = Zernikmoment(src, n, m)
# where
#   src = input image
#   n = The order of Zernike moment (scalar)
#   m = The repetition number of Zernike moment (scalar)
# and
#   Z = Complex Zernike moment
#   A = Amplitude of the moment
#   Phi = phase (angle) of the mement (in degrees)
#
# Example:
#   1- calculate the Zernike moment (n,m) for an oval shape,
#   2- rotate the oval shape around its centeroid,
#   3- calculate the Zernike moment (n,m) again,
#   4- the amplitude of the moment (A) should be the same for both images
#   5- the phase (Phi) should be equal to the angle of rotation
# ------------------------------------------------------------------------
def Zernikemoment(src, n, m):
    if src.dtype != np.float32:
        src = np.where(src > 0, 0, 1).astype(np.float32)
    if len(src.shape) == 3:
        print 'the input image src should be in gray'
        return

    H, W = src.shape
    if H > W:
        src = src[(H - W) / 2: (H + W) / 2, :]
    elif H < W:
        src = src[:, (W - H) / 2: (H + W) / 2]
```

```
    N = src.shape[0]
    if N % 2:
        src = src[:-1, :-1]
        N -= 1
    x = range(N)
    y = x
    X, Y = np.meshgrid(x, y)
    R = np.sqrt((2 * X - N + 1) ** 2 + (2 * Y - N + 1) ** 2) / N
    Theta = np.arctan2(N - 1 - 2 * Y, 2 * X - N + 1)
    R = np.where(R <= 1, 1, 0) * R

    # get the radial polynomial
    Rad = radialpoly(R, n, m)

    Product = src * Rad * np.exp(-1j * m * Theta)
    # calculate the moments
    Z = Product.sum()

    # count the number of pixels inside the unit circle
    cnt = np.count_nonzero(R) + 1
    # normalize the amplitude of moments
    Z = (n + 1) * Z / cnt
    # calculate the amplitude of the moment
    A = abs(Z)
    # calculate the phase of the mement (in degrees)
    Phi = np.angle(Z) * 180 / np.pi

    return Z, A, Phi
```

A.4 Programas para gerar os dados que foram passados ao Matlab (Classification Learner)

```
%% Prepara dados para o Classification Learner
rng('default');
clc
%% Caracteristica 1 (Edge Orientation) - Banco de Dados 1 (Amostras IFAL)
% Utilizando as amostras obtidas de trabalhos dos meus alunos no IFAL-Viçosa e edge orientation
% com segmentação da imagem de 1 até 10 partes (uma subimagem até 100 subimagens.
P = [];
N = [];
for j = 1:10
for i = 1:20

    f1 = imread(['txt',sprintf('%02d', j),'sem',sprintf('%03d', i),'.tif']);
    fneg1 = imread(['txt',sprintf('%02d', j),'ras',sprintf('%03d', i),'.tif']);
    f=imresize(f1,[200 200]); % força aspecto 1:1 e iguala dimensões
    fneg=imresize(fneg1,[200 200]); % de todas as imagens
    h = edgeOrientationHistogram(f,10,'sobel');
    hneg = edgeOrientationHistogram(fneg,10,'sobel');

     P = [P; h];
     N = [N; hneg];

end
```

```
end

%%
clc
X = [P; N];
X = svmScale(X);
Y = [ones(200,1); -1*ones(200,1)];
Z=[Y X];
A=Z;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Prepara dados para o Classification Learner
rng('default');
clc;
%% Caracteristica 2 (Hierarchical Centroid) - Banco de Dados 1 (Amostras IFAL)
% Utilizando as amostras obtidas de trabalhos dos meus alunos no
% IFAL-Viçosa e hierarchical centroid
% com profundidade de recursão de 2 até 10 (de 4 até 2044 features)
%%
P = [];
N = [];
for j = 1:10
for i = 1:20

    f1 = imread(['ex',sprintf('%02d', i),'txt',sprintf('%02d', j),'nor.tif']);
    fneg1 = imread(['ex',sprintf('%02d', i),'txt',sprintf('%02d', j),'ras.tif']);
    f=imresize(f1,[200 200]);
    fneg=imresize(fneg1,[200 200]);
    hc=hierarchicalCentroid(f,2,0);
    hcneg=hierarchicalCentroid(fneg,2,0);
    P = [P; hc];
    N = [N; hcneg];

end
end

%%
clc
X = [P; N];
X = svmScale(X);
Y = [ones(200,1); -1*ones(200,1)];
Z=[Y X];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Prepara dados para o Classification Learner
rng('default');
clc;
%% Caracteristica 3 (Momentos de Zernike) - Banco de Dados 1 (Amostras IFAL)
% Utilizando as amostras obtidas de trabalhos dos meus alunos no
% IFAL-Viçosa e momentos de Zernike
% com variação da ordem e do número de repetições.
P = [];
N = [];
for j = 1:10
for i = 1:20

    f1 = imread(['txt',sprintf('%02d', j),'sem',sprintf('%03d', i),'.tif']);
    fneg1 = imread(['txt',sprintf('%02d', j),'ras',sprintf('%03d', i),'.tif']);
    % f1 = imcomplement(imread(['txt',sprintf('%02d', j),'sem',sprintf('%03d', i),'.tif']));
    % fneg1 = imcomplement(imread(['txt',sprintf('%02d', j),'ras',sprintf('%03d', i),'.tif']));

    f=imresize(f1,[200 200]);
```

```
    fneg=imresize(fneg1,[200 200]);
    g=imbinarize(f);
    gneg=imbinarize(fneg);
    [Z, A, Phi]=Zernikmoment(g,4,10);
    [Z1, A1, Phi1]=Zernikmoment(gneg,4,10);
    % [Z2, A2, Phi2]=Zernikmoment(g,4,2);
    % [Z3, A3, Phi3]=Zernikmoment(gneg,4,2);
    zm=[real(Z) imag(Z) A Phi];
    zm1=[real(Z1) imag(Z1) A1 Phi1];
    P = [P; zm];
    N = [N; zm1];

end
end

%%
clc
X = [P; N];
X = svmScale(X);
Y = [ones(200,1); -1*ones(200,1)];
Z=[Y X];
```

Anexos

ANEXO A – Artigo Publicado na 29ª Conferência Anual Internacional do IEEE sobre Ferramentas com Inteligência Artificial (ICTAI 2017: The annual IEEE International Conference on Tools with Artificial Intelligence.

Erasure Detection and Recognition on Children's Manuscript Essays

Marcos Tenório, Evandro Costa and Tiago Vieira
Instituto de Computação
Universidade Federal de Alagoas
Av. Lourival Melo Mota s/n, Maceió/AL
Email: {marcos, evandro, tvieira}@ic.ufal.br

***Abstract*—In this paper, we present a machine learning approach to classify erasures in children's manuscript essays. We aim at highlighting challenges faced by children through their learning process, specifically when it comes to their subjective ability of converting thoughts into written text. The image processing software receives an entire scanned essay as input, then it segments each word. Afterwards, a classifier decides whether each word is "clean" or if it corresponds to an "erasure". We evaluate several Machine Learning (ML) techniques, including Support Vector Machine (SVM), boosted and bagged decision trees, k-nearest neighbor, Naive Bayes, discriminant analysis, logistic regression, and deep neural networks. To this end, manuscript essays containing erasures were scanned, each word (with and without erasure) was segmented, and were separated into two classes, providing knowledge to the system for the identification of new samples. Due to the difference between image dimensions, it was necessary to use features capable of describing the image independently of it corresponding to a letter or a word, or character dimension used by different children during writing. To solve this problem, we identify strokes directions, usually consisting of parallel lines. The obtained results show that the image descriptors when used alone achieve maximum accuracy limits of the order of 90%, but combined with the others, they exceed 98% accuracy. Compared to other used classifiers, SVM with Gaussian kernel produced higher accuracy.**

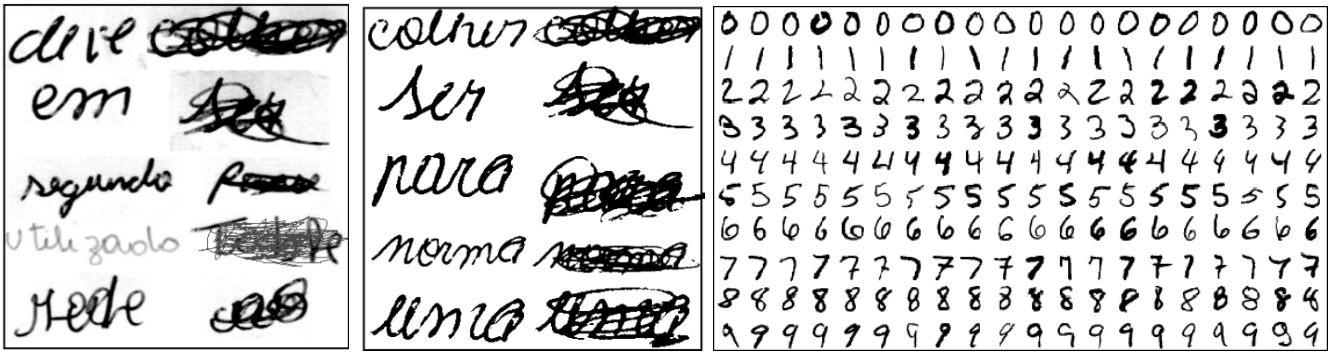
I. Introduction

Investigating the written language within the learning process is important in order to determine the main difficulties faced by students during the process of mapping spoken language into written language. During that process, students make some mistakes and then try to correct them immediately, either through a process of scratching the wrong letter / word, erasing and rewriting, or even overwriting the wrong text with the correct one, among other alternatives, markings and annotations. These attempts of corrections, called erasure, can be understood as places of conflict, that allow to visualize the writing subject in a movement of negotiation and of strangeness of its own word [1]–[3], or as an attempt of 'adjustment' that gives prominence to the bond between the spoken / oral and the written / literate and the very "constitutive heterogeneity of writing" [4], [5]. As a result of this context, in this paper, we present a machine learning approach capable of segmenting words and classifying erasures in manuscript essays. The aim of the method is to highlight challenges faced by children through their learning process, specifically when it comes to their subjective ability of converting thoughts to written text. The image processing software receives an entire scanned essay as input, segments each word and decides whether it is a "clean" word or if it corresponds to an "erasure".

The task of erasures recognition was performed by machine learning classification techniques, such as: Support Vector Machine (SVM), Naive Bayes, Neural Networks, and k-Nearest Neighbor (KNN) algorithms and we compared the performances. To accomplish this, texts with and without erasures were scanned and the words were separated into two classes, providing knowledge to the system for the identification of other samples. Due to the difference of the acquired images dimensions, it was necessary to use features capable of characterizing the image independently of it being a letter or word, or the size of the characters used in the writing. To solve this problem, we propose to identify the direction of the traces that make up the writing, because it is independent of the dimensions and characterizes well the erasures, usually consisting of parallel lines in one direction. This erasures classification can also be used through a more detailed study, the determination of the children's personality traits that lead them to commit these mistakes and can guide the teacher / instructor in using a teaching method that circumvents these deficiencies. This paper is organized as follows; Section II presents the methodology, results are reported in Section III and concluding remarks and future work are provided in Section IV.

II. Methodology

An overview of our methodology is as follows. First, we collect a database containing positive and negative samples. This is achieved by applying a morphological image processing algorithm on images of children's essays, which segments each word (or letter) by using the distance transform, where the value of each pixel becomes the Euclidean distance to the closest writing stroke. After the Distance Transform is thresholded and connected components are labeled, the result is a bounding rectangle involving each word (or letter). If an image contains an erasure, it belongs to the positive class (label $+1$), otherwise it is a negative sample (class -1). Afterwards, three features are extracted from each image, as an attempt to mathematically represent its characteristics. Once each image



(a) Examples from DB1. (b) Examples from DB2. (c) Examples from MNIST.

Fig. 1: Examples of images belonging to different databases: (a) Images from DB1, without correspondence. (b) Images from DB2. Notice the correspondence between the words. (c) Images from the publicly available dataset MNIST.

is represented by a feature vector, the database is used to train many supervised learning algorithms in order to evaluate the performances of different techniques to the problem of erasure classification. Further details are provided on each following subsection.

A. Image Databases

1) Database 1 (DB1): For the purpose of erasure recognition, we used a total of three databases. Database 1 (DB1) comprises 400 images of words and/or letters, from which 200 contain a wide variety of erasures (class $+1$) while 200 have no erasure (class -1), as shown in Fig. 1a. This initial database was obtained from the scanning of works carried out by students of a Brazilian Public School, during lectures of the Subsequent Technical Course of Informatics (STCI). There is no intentional correspondence between positive and negative classes, i.e., there is no same written word with and without erasure at the same time in the database.

2) Database 2 (DB2): Database 2 (DB2) also comprises 400 images of words and/or letters. Among these, 200 contain erasures (class $+1$) whereas 200 do not (class -1), as can be seen in Fig. 1b. It was obtained following the same procedure as DB1, except that DB2 has intentional correspondences. There are 200 images without erasures (clean) were collected and were intentionally modified, i.e., strokes were written on them to simulate erasures.

3) Database 3 (DB3): Database 3 (DB3), MNIST ("Mixed National Institute of Standards and Technology") [**?**] was constructed from the bases of NIST Special Database 3 (SD-3) and Special Database 1 (SD-1), containing binary images of handwritten digits. The National Institute of Standards and Technology (NIST) originally designated the SD-3 base as a training set and the SD-1 base as a test set. However, SD-3 is much cleaner and easier to recognize than SD-1. The reason is due to the fact that SD-3 was collected from officials of the US Census Bureau, while SD-1 was collected from high school students. Getting sensible conclusions from learning experiences requires that the result be independent of the choice of training and testing between the full set of samples. Therefore, it was necessary to build a new database by mixing the NIST datasets. SD-1 contains 58,527 images typed by 500 different writers. In contrast to SD-3, where data blocks of each writer appear in sequence, the data in SD-1 is encoded. The writer's identities for SD-1 are available and we use this information to identify the writers. Then we split SD-1 into two: (1) Characters written by the first 250 writers entered our new training set. (2) The remaining 250 writers were placed in our test suite. So we had two sets with almost 30,000 examples each. The new training set was completed using sufficient examples of SD-3, starting at number zero, to make a complete set of 60,000 training standards. Similarly, the new test set was completed with examples of SD-3 from standard 35,000 to make a complete set with 60,000 test patterns. In the experiments described here, we use only a subset of 10,000 test images (5,000 SD-1 and 5,000 SD-3).

Information about all databases are summarized in Table I.

B. Extraction of Features

After the databases are collected, the next step is extracting features from all samples as explained in the following.

1) Feature 1 (EH - Edge Orientation Histogram): The main idea is to construct a histogram with the directions of the corner gradients (edges or contours). It is possible to detect edges in an image, but we are interested in the detection of angles. This is possible through Sobel Operators. The five operators illustrated in Fig. 2 represent the strength of the gradient in these directions. The convolution on each of these masks produces a matrix of the same size as the original image indicating the gradient (force) of the edge in a particular direction. It is possible to quantify the maximum gradient in the 5 final matrices and use this to complete a histogram (*cf.*, Fig 3).

TABLE I: Summary of Databases.

Database	Number of samples	Classes
DB1	400	2
DB2	400	2
DB3	10000	10

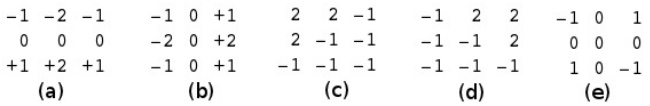


Fig. 2: Sobel masks for 5 orientations: vertical, horizontal, diagonal and non-directional

In order to avoid the amount of insignificant gradients that could potentially be introduced by this methodology, one option is to consider the edges detected by robust methods such as Canny, LoG, Prewitt, Roberts and Sobel. These detectors return an array of the same image size, with a 1 if there is an edge and 0 if there is no edge. In other words, it returns the outline of the object within the image.

Considering only the 1's, we obtain the most pronounced gradients. Since we are interested in a variable number of features (each histogram returns 5 directional features), we divide the image into parts ranging from 1 (the full image) to 10 (100 sub-images) as shown in Fig. 4.

2) Feature 2 (HC - Hierarchical Centroid): Here we calculate the hierarchical centroids of an amount of sub-images, which varies according to the depth of the function. It generates characteristics vectors with a length of 4 (four) to a depth level ranging from 2 to a maximum of 2044 for a level of 10. The general form of the number of elements of the output vector is $2 \times (2^{depth} - 2)$. In Fig. 5, we see the subdivisions of the image for the calculation of the hierarchical centroids.

3) Feature 3 (Zernike Moments): We use a function for the calculation of Zernike moments developed by [6], [7]. This function finds the Zernike moments for a binary ROI (region of interest) of size $N \times N$, where N is an even number. It returns a vector of components $[Z, A, \Phi]$, where Z is the complex moment of Zernike, A is the momentum amplitude, and Φ is the phase angle of the moment in degrees. Here, the parameters tuned in the classification were; (i) n - the order of the Zernike moment, and; (ii) m - the number of repetitions of the Zernike moments. This feature has been often used in the literature to

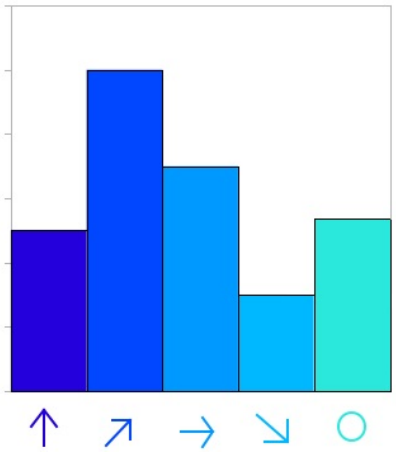
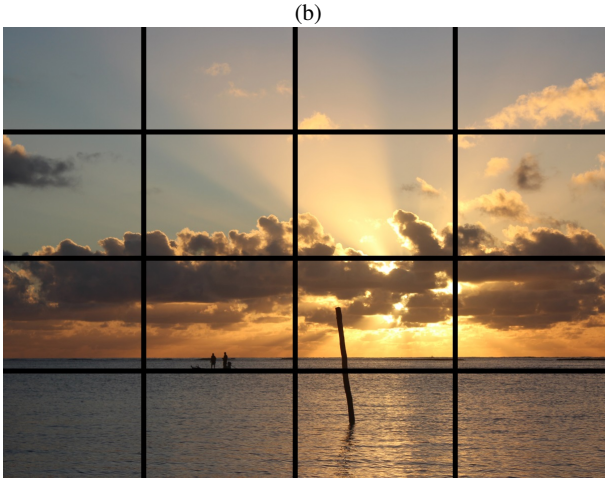


Fig. 3: Illustration of Feature 1 - Edge Orientation Histogram (EH).

(a)

(b)



(c)

Fig. 4: Examples of image sub-divisions for the computation of Edge orientation histograms. (a) Original undivided image. (b) Image divided into 3×3 regions. (c) Image divided into 4×4 parts.

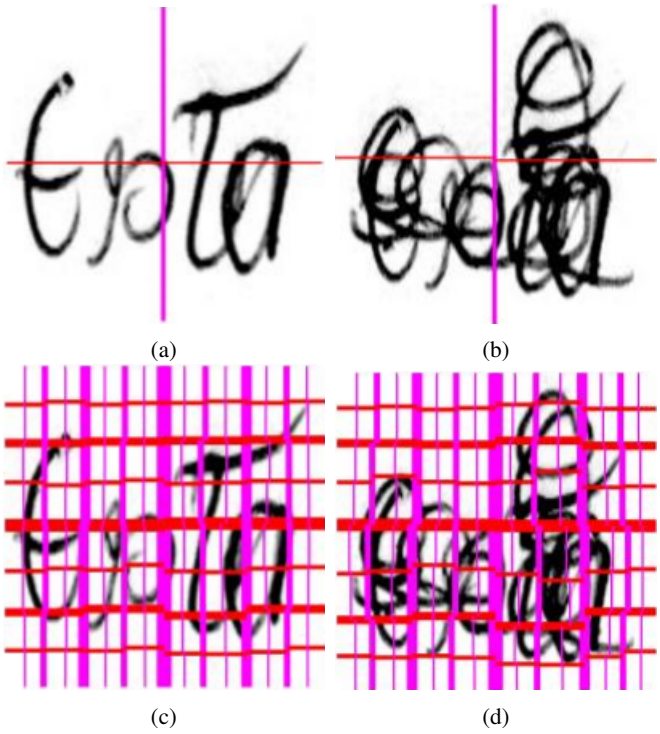


Fig. 5: Examples of images from DB2 used in the computation of the *hierarchical centroids*: (a) Negative example ($depth = 2$). (b) Positive example ($depth = 2$). (c) Negative example ($depth = 7$). (d) Positive example ($depth = 7$).

represent shapes and are invariant to rotation, as illustrated in Fig. 6.

C. Classification methods

The techniques assessed in this study are broadly used in literature. We used k-Nearest Neighbors (KNN), decision trees (TREE), Support Vector Machines (SVM) and Neural Networks (NN) in the classification of erasures. First, they are applied with "standard" parameters, i.e., the values controlling the approaches are not fine-tuned. Then we evaluated the performance of the classifiers by changing their configuration parameters, as explained in the following.

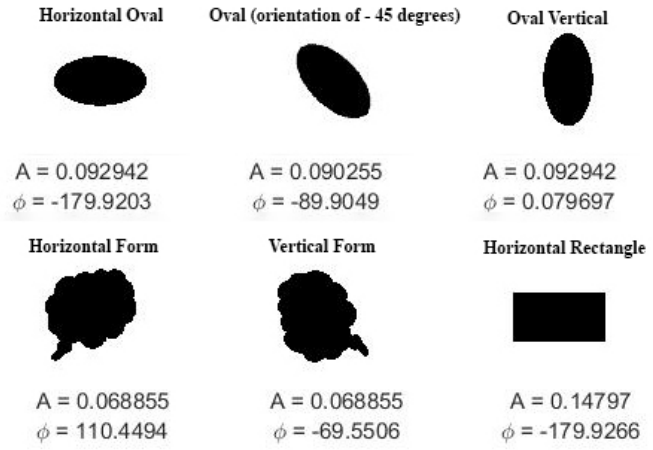


Fig. 6: Amplitudes and Phases of the Zernike Moment for $n = 4$ and $m = 2$.

1) k-Nearest Neighbors (KNN): In KNN classification, the class of a test sample is assumed to belong to the most often category among its k nearest neighbors. The performance of the model is evaluated under; (i) different distance metrics used to measure said proximity; (ii) the weights of nearest neighbors, and; (iii) the number of samples k. The number of neighbors k starts with 1 (fine) and increases to 10 (medium) and 100 (coarse).

2) Logistic Regression (LR): The goal of Logistic Regression (LR) is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent (predictor or explanatory) variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest. In our case, the categories are positive (the image contains an erasure) or negative (the image does not contain an erasure.

3) Discriminant Analysis: Discriminant analysis (DA) is a technique used to develop functions composed by the linear combination of independent variables that are capable of discerning between different categories of the dependent variable. It allows us to examine whether significant differences exist among groups, in terms of predictor variables. It also evaluates the classification accuracy. We tested different variations of discriminant analysis; Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Subspace Discriminant Analysis (SDA).

4) Support Vector Machines (SVM): A Support Vector Machine (SVM) is a classifier formally defined by a separating hyperplane. Given a set of labeled training data (supervised learning), the algorithm outputs an optimal hyperplane, capable of categorizing new samples. Besides performing linear classification, SVMs can efficiently execute non-linear classification using the kernel trick, where inputs are mapped into higher-dimensional feature spaces, where they can be (maybe) more easily separated. We tested SVMs with three different configurations, linear, polynomial (quadratic and cubic) and Gaussian with varying scales $\sigma = 1.3$ (fine), $\sigma = 5.3$ (medium) and $\sigma = 21$ (coarse).

5) Decision Trees (DT): Decision Trees are widely used to create a model that predicts the value of a target variable based on several input variables. We used many variations of trees and their parameters to evaluate their performances to tackle the recognition of erasures.

6) Ensemble models: The concept of ensemble consists in combining multiple models using the same algorithm. They are a subset of a broader set of techniques named multi-classifiers, where multiple classification methods are fused to solve a unique problem. We also used different ensembles

TABLE II: Best classifiers for Feature 1 - Edge Orientation Histogram (EH). Highest classification accuracy is 95,5%.

	Canny method				LoG kernel			
Seg./Feat.	**SVM Lin**	**SVM Quad**	**SVM Cub**	**Ens. Sub. Disc.**	**SVM Lin**	**SVM Quad**	**SVM Cub**	**Gau SVM Med**
1/5	90,30	90,00	90,00	88,30	93,50	93,30	91,80	94,00
2/20	91,00	89,50	89,00	88,80	93,30	93,50	92,00	93,80
3/45	**92,00**	91,80	90,00	91,00	**95,50**	**94,00**	93,30	91,80
4/80	91,80	92,30	90,50	**93,00**	**94,80**	**95,00**	**94,00**	92,80
5/125	92,00	92,80	**93,00**	91,30	94,00	92,30	93,00	91,80
6/180	**93,30**	91,50	92,30	92,00	93,50	93,80	92,80	92,50
7/245	**94,80**	**94,50**	93,40	91,50	**95,00**	**95,00**	**94,00**	93,00
8/320	**92,50**	**93,80**	92,30	91,30	93,50	**94,00**	**94,00**	91,80
9/405	**92,50**	91,30	90,50	87,00	**95,30**	**94,80**	93,00	90,30
10/500	90,80	**91,50**	90,80	85,50	94,00	**94,30**	93,30	90,50

	Roberts kernel				Sobel kernel			
Seg./Feat.	**Log Reg.**	**SVM Lin**	**SVM Quad**	**Gau. SVM Med**	**Log Reg**	**SVM Quad**	**SVM Cub**	**Ens. Sub. Disc.**
1/5	68,30	68,80	**77,30**	68,80	77,80	77,00	74,30	74,50
2/20	73,00	66,50	68,00	68,00	**81,50**	76,00	69,30	**79,80**
3/45	**76,50**	**75,30**	71,00	**72,50**	80,80	**79,00**	76,80	**80,30**
4/80	70,80	68,50	70,00	71,50	75,00	78,00	77,30	78,00
5/125	68,30	68,00	70,30	69,50	74,30	78,80	**79,50**	**78,50**
6/180	62,50	69,80	67,30	67,80	71,30	78,80	**79,80**	78,30
7/245	60,00	67,80	66,80	67,30	67,00	**79,80**	77,30	78,30
8/320	49,50	69,50	70,00	66,80	51,50	78,00	77,30	75,50
9/405	55,30	68,50	69,80	67,30	51,00	**79,00**	**80,50**	75,50
10/500	56,80	71,80	71,80	67,50	54,50	78,30	78,80	76,00

of the previously introduced Machine Learning techniques, namely, bagging and Bootstrap AGGregatING (Bagging).

7) Deep Neural Networks (DNN): Finally, we used Deep Learning for the classification of erasures. Specifically, we used Convolutional Neural Networks (CNN) because they are particularly useful for the interpretation of image data by automatically extracting features by adjusting the weights of internal layers.

III. Results and Discussion

In this section we present comprehensive results regarding the classification of erasures using different parameters and classifiers. Experiments outcomes are organized into three parts, as explained in the following. First we evaluate features individually, using them with different classification techniques as explained in Section II-C. Then the features are combined to assess if combinations are capable of improving the discrimination between the two classes. Finally, the images were directly input to a Convolutional Neural Network (CNN) which optimizes its weights in order to highlight discriminative features between training samples. In this case, no features are extracted from the images. During training, the CNN is capable of recognizing non-linear features more pertinent to the problem at hand.

A. Evaluating Features Individually

1) Using Feature 1 - Edge orientation Histogram (EH): The parameters modified here for an improvement in the accuracy of the classifiers were the number of sub-images (ranging from 1 to 100) and the type of filter used in the histogram calculation function (Canny, LoG, Prewitt, Roberts and Sobel). Table II shows the variation of accuracy as a function of the number of sub-images (best classifiers) and filter type (for the same amount of sub-images). The classifiers chosen for comparison were those that presented highest accuracy across the range of sub-images (1-100) and the best general classifier was used for a comparative between types of filters.

Although not necessary for the calculation of the histogram, the images were normalized to a dimension of 200×200 pixels, matching them to those used in the calculation of the other features. Table II, shows the results for the first feature (Edge Orientation Histogram - EH), which provided a highest accuracy of 95,5% for the Log of Gaussian (LoG) filter.

We use the Canny filter (upper-left part of Table II) to generate the embossed image to determine the values for the edge histogram. It provided a top accuracy of approximately 95%, using the linear classifier SVM and 245 orientations. From all tested techniques, those providing highest performances were based on SVM and Ensemble Subspace Discriminant analysis.

Using the LoG filter (upper-right region of Table II) for the generation of the embossed image to determine the values for the edge Histogram, resulted in a maximum of 95,5%, using the linear SVM classifier and 45 orientations.

In addition, the Roberts filter (bottom-left corner of Table II) was also tested for the generation of the embossed image to determine the values for the edge Histogram. A maximum of 76.5% was obtained using the Logistic Regression classifier and 45 orientations.

Finally, using the Sobel filter (bottom-right part of Table II) for the generation of the embossed image provided a maximum classification accuracy of 81,5% using Logistic Regression and 20 orientations.

2) Using Feature 2 - Hierarchical centroid: To evaluate the performance variation of Feature 2 - Hierarchical Centroid (HC) the amount of sub-images used to calculate the centroids (Table III) was tuned. The highest accuracy was 87,3% using the medium-size Gaussian kernel with Support Vector Machine (SVM) classifier.

TABLE III: Best classifiers for Feature 2 - Hierarchical Centroid (HC).

Seg./Feat.	SVM Quad	Gau. SVM Med	Ens. Bagg T
2/4	**67,80**	64,30	58,80
3/12	84,80	**87,00**	86,80
4/28	83,30	**87,00**	86,80
5/60	83,30	**87,30**	87,00
6/124	85,00	86,00	**86,50**
7/252	85,50	85,80	**86,00**
8/508	84,50	**85,80**	85,00
9/1020	**85,50**	85,30	85,30
10/2044	84,30	**86,30**	84,50

3) Using Feature 3 - Zernike Moments (ZM): For this evaluation we analyze two situations: 1) varying the order of the polynomial (2-16, even – upper-part of Table IV) and 2) the number of repetitions of the Zernike Moment (2-4, even – bottom-part of Table IV). The maximum values obtained were 72,3% and 72,8%, respectively using the Quadratic Discriminant for classification.

TABLE IV: Best classifiers for Feature 3 - Zernike moments.

Ord./Rep.	Discr Quad	Log. Reg.	SVM Lin	KNN Coa
2/2	**68,30**	**68,30**	67,80	64,50
4/2	**72,30**	71,50	72,00	68,50
6/2	**58,50**	56,00	56,30	54,50
8/2	56,00	**57,80**	55,30	54,50
10/2	54,80	53,30	52,30	**55,50**
12/2	**53,50**	**53,50**	53,30	53,30
14/2	48,50	50,20	**52,80**	49,50
16/2	**52,00**	49,80	50,20	47,00
Ord./Rep.	**Discr Quad**	**Log Reg**	**SVM Lin**	**Gau. SVM Med**
4/2	**72,80**	71,00	70,50	70,30
4/4	56,50	**57,50**	55,80	56,50

B. Combining Features

1) Combining Features 1 and 2 - Edge Orientation Histogram (EH) and Hierarchical centroid (HC): After the performance of individual features was assessed, we combined them to verify the new classification effectiveness. The best classification accuracies for all two-by-two combinations are presented in Table V. When Edge Orientation Histogram (EH) is combined with Hierarchical Centroid (HC), the maximum result achieved 97,0% for Linear Support Vector Machine (SVM).

2) Combining Features 1 and 3 - Edge Orientation Histogram (EH) with Zernike Moment (ZM): Next we combine Features 1 - Edge Orientation Histogram (EH) with Feature 3 - Zernike Moments (ZM) and results can be appreciated in Table V. The highest accuracy achieved 96,8% using the quadratic Support Vector Machine (SVM).

3) Combining Features 2 and 3 - Hierarchical centroid (HC) with Zernike Moment (ZM): The combination of the last two features, namely, Hierarchical Centroid (HC) and Zernike Moments (ZM), provides an accuracy of 89,5% as can be seen in Table V.

TABLE V: Best classifiers for different combinations of three features; (i) Edge Orientation Histogram (EH); (ii) Hierarchical Centroid (HC), and; (iii) Zernike Moments (ZM).

EH + HC			
SVM Lin	**SVM Quad**	**SVM Cub**	**Gau. SVM Med**
97,00	96,80	96,50	96,80
EH + ZM			
SVM Lin	**SVM Quad**	**SVM Cub**	**Gau. SVM Med**
95,80	**96,80**	96,00	95,80
HC + ZM			
SVM Quad	**Gau. SVM Med**	**Ens. Boo T**	**Ens. Bag. T**
89,30	86,50	85,50	**89,50**
EH + HC + ZM			
SVM Lin	**SVM Quad**	**SVM Cub**	**Gau. SVM Med**
97,00	**98,00**	**98,00**	97,00

C. Using all Features: Edge Orientation Histogram (EH) + Hierarchical centroid (HC) + Zernike moment (ZM)

Finally, we aggregated all features used to characterize the images. An increase in accuracy can be appreciated in Table V, where a Maximum value reached 98% with quadratic and cubic Support Vector Machines (SVM).

D. Deep Convolutional Neural Networks (CNN)

Finally, a Convolutional Neural Network (CNN) was applied to the problem of classifying erasures in children's essays. The result is presented in Fig. 7, which shows the increase in classification accuracy for the training and testing stages of each of the experiment. We used 80% of the database for training and 20% for testing in a Convolutional Neural Network (CNN). The topology consists of:

1) An input convolutional layer using 32 filters and linear rectification unit.
2) A MaxPooling layer responsible for down-sampling the results of the convolutions increasing the network capability of highlighting non-linear features.
3) A second convolutional layer followed by another down-sampling stage.
4) A flattening layer whose function is to concatenate all data into a vector to supply it to the next layer.
5) A densely connected layer with 100 inputs.
6) An output densely connected layer with one output.

According to our experiments, this topology provided low overfitting. The highest training and testing accuracies achieved 98% and 95%, respectively. Also, since the number of images used is low, we used a data augmentation scheme which rescales, shears, zooms and flips horizontally all images in the database before the convolution operations in the input layer.

IV. Conclusion

In this paper, we presented a supervised machine learning approach to perform the classification of erasures on children's manuscript essays. The algorithms segments each word/letter and decides whether each image contains an erasure or not by taking into account the combination of different image descriptors (representing direction, shape and intensity). We

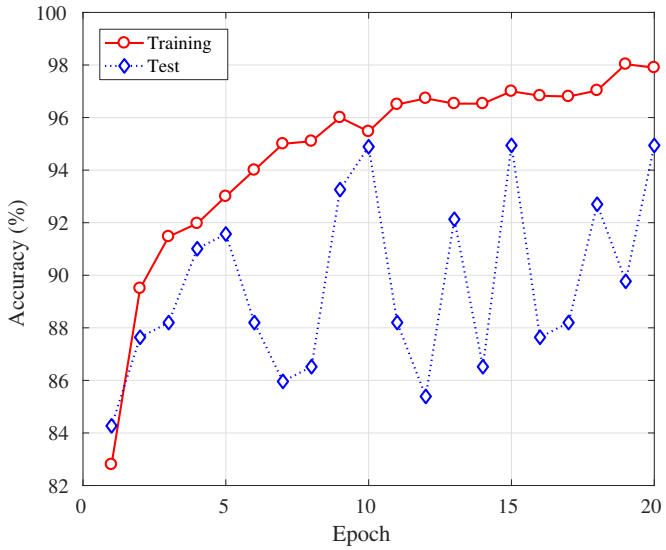


Fig. 7: Accuracies of training and validation stages for deep learning epochs. Highest training accuracy is 98%. Highest testing accuracy is 95%.

classified the erasures with many different algorithms (SVM, KNN, Gaussian SVM, etc.) and observed the evolution of the results according to the combination of the features generated by the descriptors.

It has been found that the image descriptors when used alone achieve maximum accuracy limits of the order of 95%, but, when combined with the others, they exceed 98% accuracy. The use of descriptors with different features generates characteristics that, combined, better describe the image. Indeed, individual descriptors contribute to an accuracy between 70-80% but, when combined, increase the performances to values between 96-98 %.

As immediate future work, we will focus on a larger study of erasures that can be done in groups of children with proven learning difficulties to determine specific patterns. These patterns can be used in machine learning systems as a way of determining learning disorders, such as dyslexia, dyscalculia and others, still in an early stage in the learning process.

Acknowledgment

This work has been partially supported by CNPq (grant 447336/2014-2). The authors would also like to thank the Federal Institute of Alagoas for its contributions to this work.

References

[1] E. Calil, *Escutar o invisivel: escritura & poesia na sala de aula*, ser. Coleção arte e educação. Unesp, 2008. [Online]. Available: https://books.google.com.br/books?id=nu2VPgAACAAJ

[2] C. Felipeto, *Rasura e Equívoco - no processo de escritura em sala de aula*. Eduel, 2008. [Online]. Available: https://books.google.com.br/books?id=Sx6Cw8_tJ7QC

[3] C. Capristano, "Mudanças na trajetória da criança em direção à palavra escrita," Ph.D. dissertation, Universidade Estadual de Campinas, Instituto de Estudos da Linguagem, Pós-Graduação em Lingüística Aplicada, Campinas, 2007.

[4] M. L. G. Côrrea, *O Modo Heterogêneo de Constituição da Escrita*. Martins Fontes, 2004.

[5] C. Capristano and L. Chacon, "Relações metafóricas e metonímicas: notas sobre a "aquisição" da noção de palavra," *O (In) esperado de Jakobson. Campinas: Mercado de Letras, prelo*, 2014.

[6] F. Saki, A. Tahmasbi, H. Soltanian-Zadeh, and S. B. Shokouhi, "Fast opposite weight learning rules with application in breast cancer diagnosis," *Computers in Biology and Medicine*, vol. 43, no. 1, pp. 32–41, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.compbiomed.2012.10.006

[7] A. Tahmasbi, F. Saki, and S. B. Shokouhi, "Classification of benign and malignant masses based on Zernike moments," *Computers in Biology and Medicine*, vol. 41, no. 8, pp. 726–735, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.compbiomed.2011.06.009