Trabalho 3 - Classificação de dígitos de placas de carro

Angélica Alves Viana*

Novembro de 2019

Resumo

Aplicações de classificação de objetos em imagens vêm sendo cada vez mais requisitadas em diversas áreas com destaque para numerosas aplicações na área da medicina. Por esse motivo, muitas técnicas de extração de atributos e classificadores cada vez mais robustos foram e vem sendo desenvolvidos com o objetivo de suprir essa demanda. Esse trabalho tem como objetivo apresentar, testar e analisar alguns desses classificadores e extratores de atributos na classificação de dígitos de placas de carro. Foram realizados testes com com as seguintes combinações de classificadores e extratores: Naive Bayes - LBP; Naive Bayes - Momento de Hu; Naive Bayes - GLCM; SVM - LBP; SVM - GLCM e CNN - CNN.

Palavras-chaves: classificadores. extratores de atributos. reconhecimento de dígitos.

Introdução

O reconhecimento de dígitos é uma necessidade recorrente em muitos cenários, especialmente em aplicações de reconhecimento de placas de veículos e em reconhecimento de dígitos manuscritos. De um modo geral, para que o processo de classificação de uma imagem seja possível é necessário que características essenciais da mesma sejam extraídas previamente.

Muitos extratores de atributos já foram desenvolvidos com o intuito de facilitar esse processo. Alguns deles são baseados em textura com abordagem estatística, como é o caso do GLCM, outros são baseados em formas e utilizam regiões da imagem para extrair informações de interesse, como é o caso do Momentos de Hu. Outros utilizam as fronteiras do objeto para extrair a sua forma, como descritores de Fourier e a assinatura do objeto.

Devido a essa imensa quantidade de extratores de características e classificadores existentes a tarefa de encontrar a melhor combinação para solucionar um determinado problema de classificação de imagens torna-se um procedimento difícil, por esse motivo é

*angelicaalvesviana@gmail.com

crucial destacar a importância de conhecer as características de cada extrator que será utilizado e se o vetor de atributos gerado contribui para o processo de classificação. Com o intuito de analisar o comportamento de alguns classificadores combinados com alguns extratores de características, esse trabalho apresentará os resultados da combinação dos classificadores Naive Bayes, SVM com os extratores LBP, Momentos de Hu e GLCM e a utilização de uma CNN com a finalidade de extrair os atributos e classificar.

1 Extratores de Características

1.1 Local Binary Pattern (LBP)

O *Local Binary Pattern* é um descritor de textura que foi proposto em 1996 por Ojala, Pietikäinen e Harwood cujo nome é proveniente do fato de que cada pixel da imagem é associado a um valor decimal obtido a partir da conversão de um número binário resultante da comparação dos pixels com seus vizinhos em uma vizinhança 3x3, de modo que se o valor do pixel vizinho for maior que o pixel central, este é codificado como 1, caso contrário, este é codificado como 0. Esse número binário é obtido a partir da concatenação de todos os códigos binários dentro da vizinhança, também referenciados como *Local Binary Patterns* iniciando do pixel do topo mais à esquerda seguindo um sentido horário e desconsiderando-se o pixel central. A Figura 1 ilustra esse procedimento.

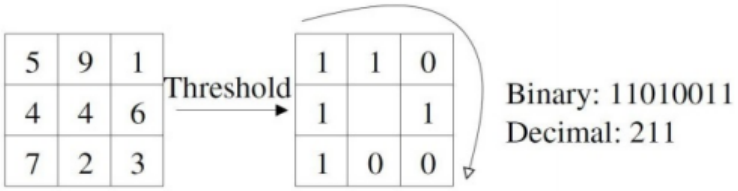

Figura 1 – Exemplo de aplicação do LBP (ZHANG, 2010).

A definição desse extrator de características dada no parágrafo anterior é baseada na ideia inicial proposta pelo autores, no entanto, percebeu-se que uma vizinhança definida de tamanho 3x3 limitava o operador de capturar características de larga escala em uma imagem, por esse motivo, posteriormente o operador LBP foi aprimorado para que fosse possível definir qualquer tamanho de vizinhança local.

Uma vizinhança local é definida como qualquer amostra de pontos espaçadas por um raio do pixel central a ser rotulado. Os pontos que não pertencem a amostra mas que estão espaçados pelo raio definido são interpolados. A estrutura do operador LBP estendido é apresentada na Figura 2.

Em Zhang (2010), a expressão matemática que define o resultado decimal do LBP é dada por

$$LBP_{P,R}(x_c, y_c) = \sum_{P=0}^{P-1} s(i_p - i_c)2^P \tag{1}$$

Onde i_c e i_p são os níveis de cinza do pixel central e de seus vizinhos respectivamente. E a função s é definida como

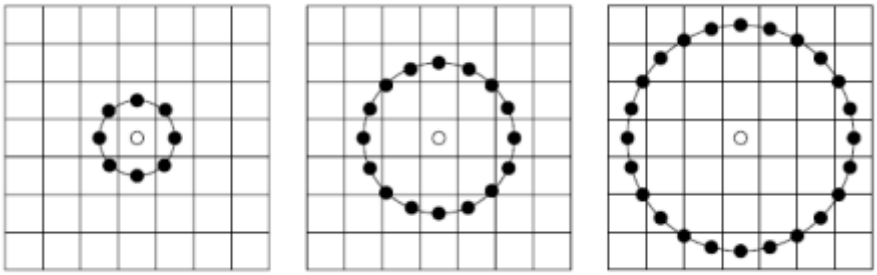

Figura 2 – Estrutura visual do LBP estendido para tamanho de raios e números de amostras distintos.

$$s = \begin{cases} 1 \ se \ x \geq 0 \\ 0 \ se \ x < 0 \end{cases} \tag{2}$$

A Figura abaixo ilustra um exemplo mostrando o passo a passo do cálculo do LBP.

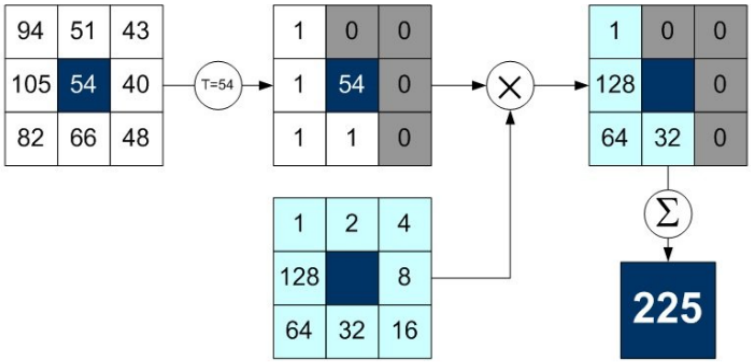

Figura 3 – Estágios do cálculo do LBP (PRAKASA, 2016).

O LBP é invariante a transformações de escala de cinza na vizinhança, preservando a ordem de intensidade. Além disso, o histograma desse operador pode ser explorado como um descritor de textura (ZHANG, 2010).

1.2 Momentos de HU

O conceito inicial de Momentos de HU foi apresentado em Hu (1962) e desde então é amplamente utilizado em análise de imagens e reconhecimento de padrões. Pode ser tido como um descritor de forma que é baseado na teoria da probabilidade, consistindo em características numéricas utilizadas para descrever variáveis randômicas (FU et al, 2018).

Hu aplicou momentos à analise de imagens com o intuito de encontrar momentos invariantes à transformações de rotação, translação e escala. Considerando-se os níveis de

cinza de cada pixel na imagem como uma função de distribuição de probabilidade $f(x, y)$, o momento geométrico de ordem (p+q) da imagem é definido como

$$m_{pq} = \int\int x^p y^q f(x, y) dx dy \;\; p\,, q \;=\; 0,\; 1\,,\; 2... \tag{3}$$

Os momentos são utilizados para descrever características de uma imagem, no entanto, eles são sensíveis a transformações de coordenadas. Diante disso, definindo o momento central de ordem (p+q), temos

$$\mu_{pq} = \int\int (x - x_0)^p (y - y_0)^q f(x, y) dx dy \;\; p,\; q \;=\; 0,\; 1,\; 2... \tag{4}$$

Onde x_0, y_0 são o centro da imagem, $x_0 = m_{10}/m_{00}$; $y_0 = m_{10}/m_{00}$. Então o momento central normalizado de ordem (p+q) de uma imagem é definido como

$$\eta_{pq} = \frac{\mu_{pq}}{{\mu_{00}}^{1+(p+q)/2}} \tag{5}$$

Desse modo, é obtido um momento invariante à translação e escala, mas ainda não é invariante a rotação. Hu conseguiu um conjunto de momentos invariantes a translação, rotação e escala a partir da combinação linear de momentos centrais de mais baixa ordem. As expressões que definem esses momentos são apresentadas abaixo

$$\begin{cases} \mathbf{C1} : \eta_{20} + \eta_{02} \\ \mathbf{C2} : (\eta_{20} - \eta_{02})^2 + 4{\eta_{11}}^2 \\ \mathbf{C3} : (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\ \mathbf{C4} : (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\ \mathbf{C5} : (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})] \\ +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{121} + \eta_{03})^2] \\ \mathbf{C6} : (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + 12)(\eta_{21} + \eta_{03}) \\ \mathbf{C7} : (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})] \\ +(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{121} + \eta_{03})^2] \end{cases} \tag{6}$$

Todos os momentos são definidos no domínio contínuo, no entanto, em processamento digital de imagens os mesmos devem estar no domínio discreto. Se a dimensão de uma imagem é dada por M x N então o momento geométrico de ordem (p+q) no domínio discreto é definido como

$$m_{pq} = \sum_{x=1}^{M} \sum_{y=1}^{N} x^p y^q f(x, y) \;\; p,\; q \;=\; 0,\; 1\,, 2... \tag{7}$$

Os momentos centrais, momentos centrais normalizados e os momentos invariantes de Hu podem ser obtidos a partir da Equação 7

1.3 Gray Level Coocurrence Matrix (GLCM)

O Gray Level Coocurence Matrix é um método de extração de características de textura estatística de segunda ordem cuja abordagem se baseia na construção de uma matriz G de dimensões L x L em que L é a quantidade de níveis de cinza presentes na imagem. O valor $P(i, j|\Delta x, \Delta y)$ é a frequência relativa com que dois pixels de intensidades i e j aparecem separados por uma distância $(\Delta x, \Delta y)$ em pixels. Uma limitação desse método de extração de características é quanto mais níveis de cinza a imagem tiver maior será a matriz G, por ser muito custoso computacionalmente, em geral, são utilizadas técnicas para redução dos níveis de cinza na imagem.

O GLCM tem sido aprimorado para ser um método popular estatístico de extração de características de textura em uma imagem. Haralick (1973) define quatorze medidas de características associadas a textura que são obtidas a partir da matriz de coocorrência G. Dessas Baraldi e Parmiggiani (1995) mostraram que apenas seis eram relevantes: Segundo momento angular, entropia, contraste, variância, correlação e homogeneidade. Essas medidas são representadas a partir da Equação 8, Equação 9, Equação 10, Equação 12, Equação 13 e Equação 14.

$$f_{sma} = \sum_{i=0}^{H_g}\sum_{i=0}^{H_g} p_{i,j}^2 \tag{8}$$

$$f_{ent} = -\sum_{i=0}^{H_g}\sum_{i=0}^{H_g} p_{i,j} log(p_{i,j}) \tag{9}$$

$$f_{con} = \sum_{i=0}^{H_g}\sum_{i=0}^{H_g} (i-j)^2 p_{i,j} \tag{10}$$

$$f_{var_i} = \sum_{i=0}^{H_g}\sum_{i=0}^{H_g} (i-\mu_i)^2 p_{i,j}^2 \tag{11}$$

$$f_{var_j} = \sum_{i=0}^{H_g}\sum_{i=0}^{H_g} (j-\mu_j)^2 p_{i,j}^2 \tag{12}$$

$$f_{corr} = \frac{1}{\sigma_x \sigma_y}\sum_{i=0}^{H_g}\sum_{i=0}^{H_g} (i-\mu_i)(j-\mu_j) p_{i,j} \tag{13}$$

$$f_{hom} = \sum_{i=0}^{H_g}\sum_{i=0}^{H_g} \frac{1}{1+(i-j)^2} p_{i,j} \tag{14}$$

2 Classificadores de Padrões

2.1 Classificador Gaussiano Naive Bayes

Em Theodoridis (2009), o classificador Naive Bayes é definido como um método robusto para classificação de dados cujo treinamento consiste em encontrar os parâmetros:

$\boldsymbol{\mu_i}$ e Σ_i, em que

$$\boldsymbol{\mu} = \sum_{k=1}^{N_i} \frac{\boldsymbol{x_k}}{N_i}$$

Onde $\boldsymbol{\mu_i}$ representa a média dos padrões da classe i e N_i o número total de padrões da classe i.

E

$$\Sigma_i = E[(\boldsymbol{x} - \boldsymbol{\mu_i})(\boldsymbol{x} - \boldsymbol{\mu_i})^T]$$

Onde Σ_i denota a matriz de covariância dos dados da classe i e E é o símbolo de esperança.

Além disso, ainda na etapa de treinamento são computadas as probabilidades a priori $P(w_i)$ de cada classe, que é dada por

$$P(\omega_i) \approx \frac{N_i}{N}$$

Em que N_i novamente denota o número de elementos da classe i e N o número total de elementos do conjunto de treinamento.

A regra de decisão do classificador é dada pelo comparativo entre as probabilidades a posteriori, que é obtida a partir da regra de bayes

$$P(\omega_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})}$$

Onde a Função Densidade de Probabilidade Gaussiana é dada por

$$P(\boldsymbol{x}) = \frac{1}{2\pi^{\frac{l}{2}}|\Sigma|^{\frac{1}{2}}} exp\left(\frac{-1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T\Sigma^{-1}(x - \boldsymbol{\mu})\right)$$

E l é um parâmetro representando a dimensionalidade dos dados.

2.2 Baseado em Discriminantes

Uma outra versão do classificador bayesiano gaussiano é realizada a partir da abordagem de discriminantes, por meio do qual são feitas algumas suposições sobre as matrizes de covariância dos dados e então são derivadas novas regras de decisão.

Primeiramente, para a implementação baseada em um discriminante quadrático, no treinamento são calculados os mesmos parâmetros encontrados no treinamento do classificador puro, ou seja, $\boldsymbol{\mu}$, Σ_i e $P(w_i)$.

A regra de decisão do classificador é dada pela fórmula abaixo:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^T{\Sigma_i}^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i}) + lnP(\omega_i) + c_i$$

Onde c_i é uma constante igual a $-\frac{l}{2}ln2\pi - \frac{1}{2}ln|\Sigma_i|$

já as regras de decisão dos classificadores bayesianos gaussianos baseados em discriminantes lineares são determinadas conforme algumas suposições:

- **Disc. Linear 1: Matrizes de covariância iguais para todas as classes:** Fazendo-se essa suposição, o discriminante linear pode ser obtido por

$$g_i(\boldsymbol{x}) = \boldsymbol{w_i}^T x + \boldsymbol{w_{i0}}$$

Onde $\boldsymbol{w_i} = \Sigma^{-1}\boldsymbol{\mu_i}$ e $\boldsymbol{w_{i0}} = lnP(\omega_i) - \frac{1}{2}\boldsymbol{\mu_i}^T\Sigma^{-1}\boldsymbol{\mu_i}$

- **Disc. Linear 2 - Matrizes de covariância Diagonal com elementos iguais:** Fazendo-se essa suposição, o discriminante linear pode ser obtido pela fórmula ainda mais simplificada

$$g_i(\boldsymbol{x}) = \frac{1}{\sigma^2}\boldsymbol{\mu_i}^T\boldsymbol{x} + \boldsymbol{w_{i0}}$$

Onde $\boldsymbol{w_{i0}} = lnP(\omega_i) - \frac{1}{2\sigma^2}\boldsymbol{\mu_i}^T\boldsymbol{\mu_i}$

- **Disc. Linear 3 - Matrizes de covariância iguais e classes equiprováveis:** Sob essas condições o compto do discriminante pode ser simplificado para

$$g_i(\boldsymbol{x}) = \frac{-1}{2}(\boldsymbol{x} - \boldsymbol{\mu_i})^T\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i})$$

- **Disc. Linear 4 - Matriz de covariância equivalente a $\sigma^2 I$:** Sob as condições de classes equiprováveis e matriz de covariância iguais, neste caso, maximizar $g_i(\boldsymbol{x})$ é o mesmo que minimizar a distância euclidiana dada por

$$d_e = ||\boldsymbol{x} - \boldsymbol{\mu_i}||$$

- **Disc. Linear 5 - Matriz de covariância não-diagonal:** Sob as condições de classes equiprováveis e matriz de covariância iguais, para este, maximizar $g_i(\boldsymbol{x})$ equivale a minimizar a distância de Mahalanobis dada por

$$d_m = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu_i})^T\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu_i})}$$

2.3 Suport Vectors Machine (SVM)

O SVM é um algoritmo supervisionado de aprendizagem de máquina desenvolvido na década de 90 cujo objetivo é encontrar um hiperplano em um espaço N-dimensional capaz de classificar os dados. É baseado em um outro algoritmo, o classificador *Maximal Margin*, que é simples e é aplicável apenas a dados linearmente separáveis.

Esse classificador é baseado na ideia de tentar maximizar a margem ao redor do hiperplano definida pelos vetores de suporte, que nada mais são do que pontos próximos e equidistantes ao hiperplano que determinam sua direção. A quantidade de vetores de suporte depende diretamente de um parâmetro de regularização C, que define a tolerância da espessura da margem.

Originalmente, o SVM foi concebido com a finalidade de apenas separar dados utilizando um modelo linear, com o intuito de superar essa limitação foi criado o conceito de *kernel*, que permite mapear um conjunto de dados de uma dimensão para uma dimensão mais elevada, de modo que seja possível traçar um hiperplano separador. Essa abordagem é baseada nas funções de *kernel*, que consistem em funções que recebem dois vetores e têm como saída um valor de produto interno. Se esse produto interno é pequeno, significa que os vetores são diferentes, caso contrário eles são bastantes similares (SUYKENS et al, 1999).

Ainda de acordo com Suykens et al (1999) existem basicamente quatro tipos de funções de kernel, são elas: *Linear*, *Polynomial*, *Radial Basis Function* (rbf) e *Sigmoid*. O *kernel* RBF que será utilizado nesse estudo pode ser tido como um transformado para novas características de mais alta dimensão a partir das medidas de distâncias entre todos os pontos de dados e um ponto específico. A função de base radial gaussiana é a mais popular e sua forma matemática é apresentada a seguir

$$k(x_i, x_j) = exp(-\gamma||x_i - x_j||^2) \tag{15}$$

Onde o parâmetro γ controla o grau de influência das novas características sobre o limite de decisão e semelhante ao parâmetro de regularização C, o γ é um parâmetro que pode ser ajustado quando se utiliza o "truque"do *kernel* (LIN e LIU, 2007).

2.4 Convolutional Neural Network (CNN)

O procedimento de seleção de um extrator de atributos adequado para uma determinada aplicação de classificação de imagens ou a seleção das métricas ideais para representar o vetor de características de uma imagem pode ser um processo exaustivo e em muitos casos se a escolha for feita erroneamente, o modelo preditor pode falhar bastante em classificar corretamente a informação desejada. Uma solução prática comumente utilizada com o intuito de evitar esse processo de extração de atributos é entregar uma imagem diretamente a uma rede neural totalmente conectada. No entanto, a desvantagem dessa solução é que nessa abordagem as informações espaciais das imagens são perdidas e todos os pixels estarão em um mesmo nível na camada de entrada da rede, não importando qualquer relação de vizinhança que exista entre eles.

Além da dificuldade já descrita relacionada a incapacidade da rede completamente conectada de captar informações referentes a distribuição espacial dos pixels, o treinamento no caso desse tipo de rede neural requer um custo de tempo relativamente grande.

Com o intuito de superar essas limitações e obter resultados muito mais significativos referentes a acurácia, em geral, utilizam-se CNN's que diferentemente de uma *fully connected* ajustam os pesos compartilhados de uma rede neural conectados a apenas algumas regiões das imagens denominados Campos Receptivos Locais. Esses pesos associados funcionam como filtros (mapas de recursos) que extraem características das imagens. isso define o procedimento de uma camada convolucional, a quantidade de camadas convolucionais em uma rede neural dependerá do problema a ser solucionado (LAWRENCE et al., 1997).

Além das camadas convolucionais, há uma camada de agrupamento responsável por reduzir a dimensionalidade dos mapas de características condensadas resultantes da aplicação dos mapas de recursos, essa camada é denominada camada de *Pooling*. Existem vários tipos de *Pooling*, em geral utiliza-se o *Max-Pooling* que gera a ativação máxima dentro de uma região de entrada 2x2.

Por fim é adicionada a rede neural uma camada de saída, cuja quantidade de neurônios está diretamente associada ao número de classes que se deseja classificar, no caso em estudo neste trabalho, por exemplo, a camada de saída conterá 10 neurônios cada um associado a um dígito. Com a arquitetura da CNN montada, após o treinamento, se uma imagem de um dígito for passada para ela, o resultado da última camada serão 10 probabilidades, cada uma associada a um dígito. A maior probabilidade estará ligada a classe que será predita. O resultado de classificação é dado a partir da aplicação de uma função de ativação à última camada.

3 Metodologia

Neste trabalho foram utilizados métodos das bibliotecas OpenCV e SkitLearn do python para extrair características de uma base de imagens de dígitos de placas de carros e classificá-los. A base utilizada nesse estudo pode ser acessada através do link https://www.dropbox.com/s/nvajalvhtqo8qvh/ocr_car_numbers_rotulado.txt?dl=0. A mesma é composta por 1226 registros e em cada linha há um número, contendo números de 0 a 9. Cada linha do conjunto de dados representa uma imagem de dimensão 35 x 35 vetorizada correspondente a um dígito seguida de um rótulo que identifica o número representado pela imagem.

A primeira etapa deste trabalho consistiu da seleção de três extratores de características de imagens. Foram escolhidos dois baseados em textura: LBP e GLCM e um baseado em formas: Momentos de Hu. Em seguida, foram selecionados dois classificadores de padrões, dos quais um deles foi o Classificador Bayesiano Gaussiano e o outro uma Rede neural totalmente conectada.

Para a realização dos testes foram feitas 20 realizações de treinamento e teste, sendo 80% dos dados separados para treinamento e 20% para teste. Para cada treinamento, foram realizadas 10 iterações no k-fold para validar a performance do modelo. As métricas de avaliação utilizadas foram a acurácia média e o desvio padrão. Além disso, ao final de todas as realizações de treinamento foi apresentada a matriz de confusão média do melhor modelo encontrado.

Os testes realizados foram resultantes ds seguintes combinações: Naive Bayes - LBP; Naive Bayes - Momentos de Hu; Naive Bayes - GLCM; SVM - LBP; SVM - GLCM e CNN - CNN, cujos resultados são apresentados e analisados na próxima seção.

A arquitetura da rede neural convolucional utilizada foi definida como uma camada de convolução utilizando 64 mapas de recursos de tamanho 3x3 aplicados a imagens de tamanho 35 x 35. Na sequência foi adicionada uma camada de *pooling* utilizando um *max-pooling* de tamanho 2x2. Em seguida, a saída da camada de *pooling* será uma entrada para uma rede neural completamente conectada com 128 neurônios e uma função de ativação "relu", já na camada de saída são definidos 10 neurônios cada um referente a um dígito e uma função de ativação "softmax". O otimizador utilizado foi o "Adam". O treinamento da CNN foi dado em 10 épocas.

4 Análise e Interpretação dos Resultados

Os resultados de classificação obtidos com a aplicação dos extratores de características e os classificadores apresentados nas seções anteriores são descritos e discutidos a seguir.

O primeiro experimento realizado foi utilizando a combinação do extrator de características LBP e o Classificador Naive Bayes. Os parâmetros do LBP foram definidos como $numPoints = 8$ e $radius = 3$ por terem sido os valores que proporcionaram um maior valor de acurácia tanto no treinamento como no teste dentre os que foram testados. Os resultados de acurácia média, desvio padrão médio tanto no treinamento como no teste são apresentados abaixo.

As matrizes de confusão médias tanto do treinamento como do teste são apresentadas abaixo

Tabela 1 – Resultados de Acurácia e Desvio Padrão Médios - LBP e NB

	A. M.	**D. P. M.**
Treinamento	$90,20$	$1,11$ x 10^{-16}
Teste	$85,36$	$1,11$ x 10^{-16}

Tabela 2 – Matriz de confusão média - Treinamento e Teste (LBP e NB)

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	8	0	0	0	0	0	0	0	0	1
Class 1	0	14	0	0	0	0	0	0	0	0
Class 2	0	0	5	1	0	1	0	0	0	0
Class 3	0	1	1	10	0	1	0	0	0	0
Class 4	0	0	1	0	15	0	0	1	0	0
Class 5	0	0	1	1	0	12	0	0	0	0
Class 6	0	0	0	0	0	1	8	0	0	2
Class 7	0	0	0	0	0	0	0	6	0	1
Class 8	1	0	0	0	1	0	0	0	7	0
Class 9	0	0	0	1	0	0	4	0	0	5

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	22	0	0	0	0	0	0	0	0	1
Class 1	0	25	0	1	0	0	0	0	0	0
Class 2	0	0	15	1	0	1	0	0	0	0
Class 3	0	0	1	27	0	1	0	0	0	0
Class 4	1	0	1	0	35	0	0	1	0	0
Class 5	0	0	5	3	0	21	0	0	0	0
Class 6	0	0	0	0	0	0	21	0	0	4
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	3	0	0	0	0	0	0	0	16	0
Class 9	0	0	0	1	0	0	13	0	0	10

A partir de uma análise rápida das matrizes de confusão médias do treinamento e do teste do classificador Naive Bayes para classificação dos dígitos, é possível observar que a maior taxa de erro do modelo está relacionada a predição dos números 6 e 9, o método prediz o 9 sendo o 6 e vice-versa. O que faz sentido, já que o LBP originalmente não é um método invariante a rotação e esses dígitos podem ser tidos como uma rotação de 180° um do outro.

O segundo experimento foi realizado com a combinação do extrator de características Momentos de Hu e novamente o classificador Naive Bayes. Os resultados de acurácia e desvio padrão médios obtidos das 20 realizações de treinamento e teste podem ser visualizados na

tabela abaixo

Tabela 3 – Resultados de Acurácia e Desvio Padrão Médios - MH e NB

	A. M.	**D. P. M.**
Treinamento	78,87	$1,11 \text{ x } 10^{-16}$
Teste	75,60	0,00

As matrizes de confusão médias dos treinamentos e dos testes das 20 realizações são apresentadas abaixo

Tabela 4 – Matriz de confusão média - Treinamento e Teste (MH e NB)

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	7	0	0	0	0	1	1	0	1	0
Class 1	0	14	0	0	0	0	0	1	0	0
Class 2	0	0	6	0	0	1	0	0	0	0
Class 3	0	1	1	11	0	0	0	0	0	0
Class 4	0	0	0	0	16	0	0	0	0	0
Class 5	0	0	1	0	0	12	1	0	0	0
Class 6	2	0	0	0	0	1	8	0	0	1
Class 7	0	0	0	0	0	0	0	7	0	0
Class 8	6	0	0	0	1	1	2	0	1	0
Class 9	2	0	0	0	1	1	6	0	0	1

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	16	0	0	0	0	2	4	0	0	0
Class 1	0	26	0	0	0	0	0	0	0	0
Class 2	0	0	15	1	0	0	0	0	0	0
Class 3	0	0	0	29	0	0	0	0	0	0
Class 4	1	0	0	0	37	0	0	0	0	0
Class 5	0	0	1	0	0	23	5	0	0	0
Class 6	1	0	0	0	0	2	22	0	0	0
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	17	0	0	0	0	0	2	0	0	0
Class 9	4	0	0	0	0	3	17	0	0	0

Novamente o classificador "confunde"bastante o dígito "9"com o "6", no entanto classifica bem o dígito "6". Já o dígito "8"o classificador erra todos classificando-o como "0"em quase todos os casos. Os demais dígitos são quase sempre classificados corretamente. Os erros cometidos pelo modelo são justificados facilmente pela funcionalidade do extrator de características utilizado, que é baseado em formas, como o dígito 8 é semelhante ao dígito 0 em sua forma é razoável conceber essa "falha"do modelo.

O terceiro experimento consistiu em novamente utilizar o classificador Naive Bayes, no entanto, agora conjuntamente com o extrator de características GLCM utilizando como descritores o Segundo momento angular, a dissimilaridade, o contraste, a energia, a correlação e a homogeneidade. Medidas essas disponíveis como parâmetros para a função *greycroprops* da biblioteca *skimage*. Os valores dos parâmetros Δx e Δy definidos na seção 1.3 foram fixados como sendo 1 e 0 respectivamente após uma busca exaustiva. Abaixo são apresentados os resultados de acurácia e desvio padrão médios obtidos das 20 realizações de treinamento e teste

Tabela 5 – Resultados de Acurácia e Desvio Padrão Médios - MH e NB

	A. M.	**D. P. M.**
Treinamento	73, 77	0, 00
Teste	72, 76	$1,11 \text{ x } 10^{-16}$

As matrizes de confusão médias dos 20 treinamentos e testes realizados são apresentadas a seguir

Tabela 6 – Matrizes de confusão média - Treinamento e Teste (GLCM e NB)

Predicted

Actual	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	8	0	0	0	0	0	0	0	1	0
Class 1	0	14	0	0	0	0	0	1	0	0
Class 2	0	0	4	1	0	2	0	1	0	0
Class 3	0	1	1	5	3	4	1	0	0	0
Class 4	0	0	0	2	14	1	1	0	0	0
Class 5	0	0	5	2	2	6	1	1	0	0
Class 6	0	0	0	0	1	0	7	0	1	3
Class 7	0	0	0	1	0	0	0	6	0	0
Class 8	0	0	0	0	1	0	0	0	8	0
Class 9	0	0	0	0	1	0	4	0	1	4

Predicted

Actual	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	22	0	0	0	0	0	0	0	0	0
Class 1	0	26	0	0	0	0	0	0	0	0
Class 2	0	0	11	1	0	3	0	1	0	0
Class 3	0	0	0	15	6	7	1	0	0	0
Class 4	1	0	0	6	30	0	1	0	0	0
Class 5	0	0	10	5	2	10	2	0	0	0
Class 6	0	0	0	0	0	0	20	0	0	5
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	0	0	0	0	0	0	2	0	19	0
Class 9	0	0	0	0	1	0	14	0	1	8

Utilizando o extrator GLCM novamente o classificador teve grande dificuldade em classificar o dígito "9", cometendo mais erros do que acertos. Diferente dos testes envolvendo os outros dois extratores, em que o modelo classificou razoavelmente bem as demais classes, com o GLCM o modelo erra bastante na predição de outras classes, como as classes "3", "4", "5"e "6". Essa característica está diretamente associada as medidas utilizadas na construção do vetor de características da imagem, que evidentemente não foram boas o suficiente para permitir uma classificação com um alto valor de acurácia para todas as classes.

a segunda etapa de experimentos foi realizada com o classificador SVM com *kernel* RBF. Alguns valores para o parâmetro C foram testados e o que resultou em melhores valores de acurácia tanto no treinamento como no teste foi C = 300. Os resultados referentes a acurácia e desvio padrão médios das 20 realizações são apresentados na tabela abaixo

Tabela 7 – Resultados de Acurácia e Desvio Padrão Médios - MH e NB

	A. M.	**D. P. M.**
Treinamento	$87,87$	$0,00$
Teste	$86,58$	$0,00$

As matrizes de confusão médias referente as 20 realizações de treinamento e teste são apresentadas a seguir

Tabela 8 – Matrizes de confusão média - Treinamento e Teste (LBP e SVM)

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	8	0	0	0	0	0	0	0	0	1
Class 1	0	14	0	0	0	0	0	0	0	0
Class 2	0	0	1	5	0	1	0	0	0	0
Class 3	0	1	0	11	0	0	0	0	0	0
Class 4	0	0	0	0	16	0	0	1	0	0
Class 5	0	0	1	1	0	13	0	0	0	0
Class 6	0	0	0	0	0	0	9	0	0	2
Class 7	0	0	0	0	0	1	0	6	0	0
Class 8	0	0	0	0	1	0	0	0	8	0
Class 9	0	0	0	0	0	0	4	0	0	5

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	22	0	0	0	0	0	0	0	0	0
Class 1	0	26	0	0	0	0	0	0	0	0
Class 2	0	0	3	12	0	1	0	0	0	0
Class 3	0	0	0	29	0	0	0	0	0	0
Class 4	1	0	0	1	36	0	0	0	0	0
Class 5	0	0	1	2	0	26	0	0	0	0
Class 6	0	0	0	0	0	0	22	0	0	3
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	0	0	0	0	0	0	0	0	19	0
Class 9	0	0	0	0	0	0	12	0	0	12

Como é possível observar o modelo treinado classifica bem quase todos os números exceto os números "2"e "9", em que a quantidade de erros cometidos é inaceitável. O "9"é classificado como "6"pela propriedade do LBP de não apresentar invariância a rotação, como já foi mencionado anteriormente.

O segundo experimento realizado com o classificador SVM foi gerado a partir da combinação do mesmo com o extrator de características GLCM. os resultados de acurácia e desvio padrão médios são apresentados na tabela abaixo.

Tabela 9 – Resultados de Acurácia e Desvio Padrão Médios - GLCM e SVM

	A. M.	**D. P. M.**
Treinamento	$67,24$	$1,11 \text{ x } 10^{-16}$
Teste	$66,26$	$1,11 \text{ x } 10^{-16}$

Abaixo são apresentadas as matrizes de confusão médias dos treinamentos e dos testes das 20 realizações

Tabela 10 – Matrizes de confusão média - Treinamento e Teste (GLCM e SVM)

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	8	0	0	0	0	0	0	0	1	0
Class 1	0	14	0	0	0	0	0	0	0	0
Class 2	0	0	0	0	1	6	0	0	0	0
Class 3	0	1	0	0	6	6	1	0	0	0
Class 4	0	0	0	0	14	2	0	0	0	0
Class 5	0	0	0	0	4	9	0	0	0	0
Class 6	0	0	0	0	1	0	10	0	1	0
Class 7	0	1	0	0	1	0	0	6	0	0
Class 8	1	0	0	0	1	0	0	0	7	0
Class 9	0	0	0	0	1	0	9	0	1	0

Actual \ Predicted	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	22	0	0	0	0	0	0	0	0	0
Class 1	0	26	0	0	0	0	0	0	0	0
Class 2	0	0	0	0	1	15	0	0	0	0
Class 3	0	0	0	0	16	12	1	0	0	0
Class 4	1	0	0	0	33	4	0	0	0	0
Class 5	0	0	0	0	7	20	2	0	0	0
Class 6	0	0	0	0	0	0	25	0	0	0
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	0	0	0	0	0	0	0	0	19	0
Class 9	0	0	0	0	1	0	23	0	0	0

Observa-se nesse caso que o classificador apenas é bom em classificar cinco dígitos: "0", "1", "6", "7"e "8". Os demais são bastante confundidos, os dígitos "2"e "3"não foram aprendidos. E novamente o dígito "9"é classificado como sendo "6".

Por fim, foi realizado um experimento utilizando uma Rede Neural Convolucional cujas especificações são apresentadas na seção anterior. Os resultados de acurácia e desvio padrão médios são apresentados na tabela a seguir

Tabela 11 – Resultados de Acurácia e Desvio Padrão Médios - GLCM e SVM

	A. M.	**D. P. M.**
Treinamento	$99,19$	$1,11 \text{ x } 10^{-16}$
Teste	$99,18$	$1,11 \text{ x } 10^{-16}$

As matrizes de confusão médias de treinamento e teste são apresentadas abaixo

Tabela 12 – Matrizes de confusão média - Treinamento e Teste (CNN)

Actual 0	Predicted									
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	9	0	0	0	0	0	0	0	0	0
Class 1	0	14	0	0	0	0	0	0	0	0
Class 2	0	0	7	0	0		0	0	0	0
Class 3	0	0	0	14	0	0	0	0	0	0
Class 4	0	0	0	0	16	0	0	0	0	0
Class 5	0	0	0	0	0	13	0	0	0	0
Class 6	0	0	0	0	0	0	12	0	0	0
Class 7	0	0	0	0	0	0	0	8	0	0
Class 8	0	0	0	0	0	0	0	0	9	0
Class 9	0	0	0	0	0	0	0	0	0	11

Actual	Predicted									
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	22	0	0	0	0	0	0	0	0	0
Class 1	0	26	0	0	0	0	0	0	0	0
Class 2	0	0	16	0	0	0	0	0	0	0
Class 3	0	0	0	29	0	0	0	0	0	0
Class 4	1	0	0	0	37	0	0	0	0	0
Class 5	0	0	0	0	0	29	0	0	0	0
Class 6	0	0	0	0	0	0	25	0	0	0
Class 7	0	0	0	0	0	0	0	18	0	0
Class 8	0	0	0	0	0	0	0	0	19	0
Class 9	0	0	0	0	0	0	0	0	0	24

Esses resultados permitem inferir que as redes neurais convolucionais são robustas para fins de extração de características e classificação de imagens, de modo que os valores de acurácia encontrados com esse modelo se destacaram com relação aos demais classificadores utilizados neste estudo conjuntamente com os outros extratores de características. Isso se deve ao fato de as redes neurais convolucionais serem capazes de aprender os filtros extratores de características ideais para cada problema a ser solucionado, cabendo ao programador apenas ajustar corretamente os parâmetros das redes.

5 Conclusão

A partir da análise dos resultados obtidos foi possível perceber que a melhor combinação de um extrator de atributos com um classificador que não fosse baseada em `Deep Learning` foi a escolha do Naive Bayes com o LBP, obtendo resultados superiores quando comparadas as demais combinações. Isso se deve ao fato de esse extrator utilizar

uma medida baseada na vizinhança de cada pixel da imagem. Observou-se também que em todos os casos de combinações com algoritmos de aprendizagem de máquina, houveram bastantes erros na classificação do dígito "9", sendo constantemente classificado como "6", fato que não ocorreu com a classificação utilizando CNN, isso porque como mencionado na seção 2.4 as CNN's são invariantes a rotação. A performance dos demais classificadores foram prejudicadas essencialmente pela escolha dos algoritmos de extração de atributos, que não são robustos o suficiente para reconhecer variações de rotação e até mesmo de forma dos dígitos.

6 Referências

BARALDI, A.; PARMIGGIANI, F. **An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters**.. IEEE Transactions on Geoscience and Remote Sensing, 33: 293–304, 1995.

Fu, Z.L.; et al. **A Moment-Based Shape Similarity Measurement for Areal Entities in Geographical Vector Data**. ISPRS Int. J. Geo Inf. 2018, 7, 1–19.

LIN, S. L.; LIU, Z. **Parameter selection in SVM with RBF kernel function**. J. Zhejiang Univ. Technol. 2007, 35, 163.

LAWRENCE, S., et al . **Face recognition:** A convolutional neural-network approach. IEEE Trans. Neural Netw. 8, 98–113, 1997.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D.. **A Comparative Study of Texture Measures with Classification Based on Feature Distributions**, Pattern Recognition, vol. 29, pp. 51-59, 1996.

NAGI, J., et al. **Max-pooling convolutional neural networks for vision-based hand gesture recognition**. In Signal and Image Processing Applications, 2011 IEEE International Conference on. 342–347, 2011.

PRAKASA, E. **Ekstraksi Ciri Tekstur dengan Menggunakan Local Binary Pattern Texture Feature Extraction by Using Local Binary Pattern**, vol. 9, no. 2, pp. 45-48, 2016.

SUYKENS, J. A. K.; et al. **Least squares support vector machine classifiers: a large scale algorithm**. In Proceedings of the European Conference on Circuit Theory and Design, 1999.

THEODORIDIS, S; KOUTROUMBAS, K. **Pattern Recognition**. Academic Press, USA, 4th edition, 2009.

HARALICK, R.M.; SHANMUGAM, K.; DINSTEIN, I.. **Texture Features for Image Classification**, IEEE Trans. Systems, Man, and Cybernetics, vol. 3, no. 6, pp. 610-621, 1973.

HU, M. **Visual Pattern Recognition by Moment Invariants**. IRE Transactions on Information Theory, IT-08, 179-187, 1962.

ZHANG, J.; et al. **Boosted local structured hog-lbp for object localization**. In CVPR, 2010.