http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2021.v13.n1.e352
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 03/03/2021 Revisado: 24/05/2021 Aceito: 03/08/2021

RECONHECIMENTO DE CÉDULAS DO REAL A PARTIR DE IMAGENS USANDO CNN PARA AUXILIAR DEFICIENTES VISUAIS

REAL BANKNOTES RECOGNITION FROM IMAGES USING CNN TO ASSIST THE VISUALLY IMPAIRED

Alisson Pereira Anjos[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Danillo Roberto Pereira[1], Mário Augusto Pazoti[1], Almir Olivette Artero[2], Marco Antonio Piteri[2]

[1]Faculdade de Informática de Presidente Prudente, Unoeste - Universidade do Oeste Paulista, Presidente Prudente
alissonerdx@outlook.com, chico@unoeste.br, llalmeida@unoeste.br, danilopereira@unoeste.br, mario@unoeste.br
[2]Faculdade de Ciências e Tecnologia, UNESP - Universidade Estadual Paulista
Departamento de Matemática de Computação, Presidente Prudente
almir.artero@unesp.br, marco.piteri@unesp.br

RESUMO – O reconhecimento de cédulas de Real através do toque sempre foi um problema encontrado por deficientes visuais. O avanço da tecnologia torna possível resolver este problema computacionalmente. Neste trabalho, é apresentado um método para realizar o reconhecimento de cédulas de Real a partir de imagens utilizando algoritmos de visão computacional e inteligência artificial. Os resultados mostram que o custo computacional e a taxa de reconhecimento são aceitáveis para uso em ambientes não controlados. O tempo de processamento para o reconhecimento de cada cédula do Real foi de 200 milissegundos, com acurácia de 91,67%.
Palavras-chave: Reconhecimento de cédulas monetárias, CNN, Rede Neural, Cédulas monetárias Brasileiras.

ABSTRACT – Real Banknotes recognition through touch has always been a problem found by visually impaired. The advancement of technology makes it possible to solve this problem computationally. In this work, we present a method to perform Real Banknotes recognition from images using computational vision and artificial intelligence algorithms. The results show that computational cost and recognition rate are acceptable for use in uncontrolled environments. The processing time for recognition of each Real banknote was 200 milliseconds, with an accuracy of 91.67%.
Keywords: Banknote recognition, CNN, Neural Network, Brazilian Banknote.

1. INTRODUÇÃO

No Brasil em 2017 existia uma população de mais de seis milhões de deficientes visuais (ESTATÍSTICAS DA DEFICIÊNCIA VISUAL, 2010). Muitos deles ainda encontram problemas de acessibilidade, como, por exemplo, dificuldades ao caminhar pelas ruas, realizar compras e utilizar recursos tecnológicos (computadores, celulares, tablets, etc.) (VEQUETINE; ZANCHETTA; BRAGA, 2013). Algumas das dificuldades encontradas pelos deficientes visuais podem ser diminuídas por meio do uso da tecnologia, como

utilizar computadores para ler jornais, realizar pesquisas acadêmicas, entre outras (SÁ, 2006).

Dentro das dificuldades existentes está um problema bastante simples, o reconhecimento de cédulas monetárias do Real através do tato. Cédulas monetárias ainda são o meio mais comum de pagamento e a maneira mais usada para a realização de transações pessoais (MOMBACH, 2015).

Os deficientes visuais brasileiros reconhecem as cédulas do Real por meio do tato ou com o auxílio de outras pessoas, pois eles, não possuem métodos mais eficientes de realizar tal procedimento. No entanto, nem sempre os relevos estão presentes nas superfícies das cédulas devido aos desgastes que sofrem com a frequente manipulação e fatores naturais (COMO ..., 2009).

Os idosos possuem uma dificuldade maior, a idade avançada prejudica a sensibilidade, e com isso, torna-se difícil notar os relevos existentes nas superfícies das cédulas, sendo assim, eles acabam optando por solicitar a ajuda de outras pessoas (MOMBACH, 2015).

Existem muitos desafios a serem considerados ao realizar a identificação de cédulas do Real utilizando imagens, como, por exemplo, cédulas amassadas, rasgadas, dobradas, imagem com baixa iluminação, diferentes pontos de vista de captura, poluição visual no cenário (muitos objetos de fundo), qualidade da imagem, oclusão parcial, entre outros.

Abordagens que apenas utilizam visão computacional (GHELLERE, 2015), muitas vezes, não são suficientes para atingir um resultado relevante no reconhecimento de cédulas a partir de imagens, principalmente quando se deparam com os desafios mencionados. Ao contrário das cédulas do Dólar e do Euro, por exemplo, as cédulas do Real possuem muitas características semelhantes, principalmente no lado que contém o símbolo da Efígie da República, como mostrado na Figura 1. Essas características são semelhantes em todas as notas e representam um problema quando se deseja realizar a diferenciação entre as cédulas de cada valor. A dificuldade aumenta principalmente quando a imagem não possui a região da cédula com o número correspondente ao valor ou a gravura do animal.

Figura 1. Cédulas do Real em ambos os lados.

Fonte: Autor (2019).

Diante dos desafios e dificuldades expostas, este trabalho tem como objetivo utilizar técnicas de inteligência artificial e visão computacional para realizar o reconhecimento do valor de cada cédula do Real (notas de 2, 5, 10, 20, 50 e 100). Para realizar o reconhecimento das cédulas do Real foi utilizada uma rede neural convolucional que possibilita obter um alto nível de abstração no reconhecimento de objetos em imagens. Através do uso desta tecnologia é possível realizar o reconhecimento independente de tamanho, de características de cores, de posicionamento do objeto na imagem, de iluminação e de diferentes pontos de vista. Por esses motivos, o uso de uma rede neural convolucional foi de grande importância neste trabalho (OLIVEIRA, 2017).

O trabalho está organizado da seguinte maneira. Na Seção 2 são descritos os conceitos a respeito de Redes Neurais Convolucionais. Na Seção 3 é apresentado o método proposto para resolver o problema de reconhecer cédulas do Real a partir de imagens usando CNN. Por fim, na Seção 4 são feitas as considerações finais e propostas de trabalhos futuros.

2. REDES NEURAIS CONVOLUCIONAIS

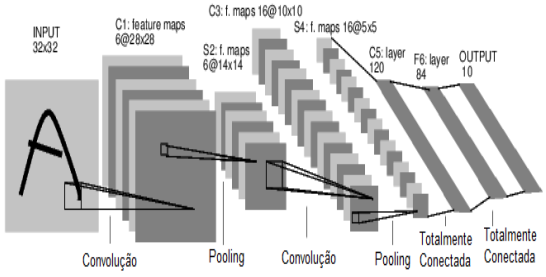
Uma rede neural convolucional (*Convolutional Neural Network* - CNN) é uma classe de redes neurais artificiais do tipo *feed-forward* que vem sendo aplicada com sucesso no processamento e análise de imagens digitais. De acordo com LeCun et al. (2015), as CNNs são redes multicamadas projetadas para o reconhecimento de padrões diretamente dos

pixels sem a necessidade de um pré-processamento complexo. Isso significa que a rede "aprende" os filtros que em um algoritmo tradicional precisariam ser implementados manualmente. Elas conseguem reconhecer padrões extremamente complexos e são adaptáveis a distorções e transformações geométricas. É exatamente essa independência de um conhecimento específico e do esforço humano no desenvolvimento de suas funcionalidades básicas a maior vantagem de sua aplicação (ARAÚJO et al., 2017) (GUO et al., 2017) (YANG; LI, 2017) (SILVA et al., 2017).

A LeNet (LECUN et al., 1989), proposta por Yann LeCun em 1989, foi um dos primeiros projetos de CNNs, tendo a mesma auxiliado a impulsionar o campo de *Deep Learning*. Inicialmente, foi utilizada para reconhecimento de caracteres, tais como código postal e dígitos numéricos. Novas arquiteturas foram propostas nos últimos anos como forma de melhoria da LeNet, embora as versões de CNNs melhoradas compartilhem de conceitos fundamentais.

As CNNs são formadas por sequências de camadas e cada uma destas possui uma função específica na propagação do sinal de entrada. A Figura 2 ilustra a arquitetura de uma LeNet e suas três principais camadas: convolucionais, de *pooling* e totalmente conectadas.

Figura 2. Ilustração da arquitetura de uma LeNet que classifica imagens de entradas em caracteres e suas três principais camadas: convolucionais, de *pooling* e totalmente conectadas.



Fonte: (LECUN et al., 1998).

As camadas convolucionais são responsáveis por extrair atributos dos volumes de entradas. As camadas de *pooling* são responsáveis por reduzir a dimensão espacial resultante após a execução das camadas convolucionais, e ajudam a tornar a representação invariante a pequenas translações na entrada. As camadas totalmente conectadas são responsáveis pela propagação do sinal por meio da multiplicação ponto a ponto e o uso de uma função de ativação. A saída da CNN é a probabilidade de a imagem de entrada pertencer a uma das classes para qual a rede foi treinada (WU, 2017).

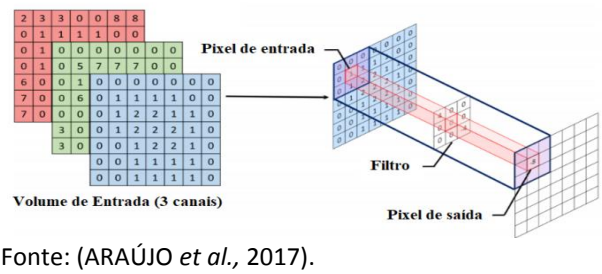
2.1. Camada Convolucional

As camadas convolucionais consistem de um conjunto de filtros que recebem como entrada um arranjo 3D, também chamado de volume. Cada filtro possui dimensão reduzida, porém, ele se estende por toda a profundidade do volume de entrada. Por exemplo, se a imagem for colorida, então ela possui 3 canais (R, G, B) e o filtro da primeira camada convolucional terá tamanho 5x5x3 (5 pixels de altura e largura, e profundidade igual a 3). Automaticamente, durante o processo de treinamento da rede, esses filtros são ajustados para que sejam ativados em presença de características relevantes identificadas no volume de entrada, como orientação de bordas e cores (KARPATHY, 2019). A relevância é avaliada de tal forma que os resultados sejam otimizados em função de um conjunto de amostras previamente classificadas.

Cada um desses filtros dá origem a uma estrutura conectada localmente que percorre toda a extensão do volume de entrada. O produto escalar entre os valores de um filtro e cada posição do volume de entrada é uma operação conhecida como convolução, a qual é ilustrada na Figura 3. Os valores resultantes após a operação de convolução passam por uma função de ativação, e a mais comum é a função ReLU (*Rectified Linear Units*) (IDE; KURITA, 2017). A principal vantagem da utilização da função de ativação ReLU é que os neurônios não são ativados todos ao mesmo tempo. Isto significa que quando se tem uma entrada em que o valor é negativo, a saída correspondente a esta entrada será convertida para zero e o neurônio não será ativado (AGARAP, 2018). Essa função pode ser calculada pela Equação (1).

$$f(x) = \max(0, x) \qquad (1)$$

Figura 3. Ilustração de convolução entre um filtro 3x3 e o volume de entrada.

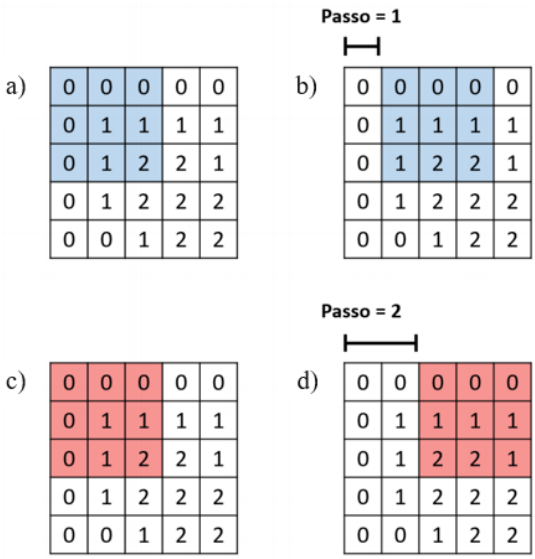


Fonte: (ARAÚJO *et al.,* 2017).

Existem três parâmetros que controlam o tamanho do volume resultante da camada convolucional: profundidade (*depth*), passo (*stride*) e zero-*padding*. A profundidade do volume resultante é igual ao número de filtros utilizados. Cada um desses filtros será responsável por extrair características diferentes no volume de entrada. Portanto, quanto maior o número de filtros maior o número de características extraídas, consequentemente a complexidade computacional, relativa ao tempo e ao uso de memória, também será maior (ARAÚJO et al., 2017).

Enquanto a profundidade do volume resultante depende somente do número de filtros utilizados, a altura e largura do volume resultante dependem do passo (*stride*) e do zero-*padding*. O parâmetro "passo" especifica o tamanho do salto na operação de convolução, como ilustrado na Figura 4. Quando o passo é igual a 1, o filtro salta somente uma posição por vez. Quando o passo é igual a 2, o filtro salta duas posições por vez. Quanto maior o valor do passo, menor será a altura e largura do volume resultante, porém, características importantes podem ser perdidas. Por esse motivo, é incomum se utilizar o valor de salto maior que 2. O zero-*padding* é uma técnica que permite preservar o tamanho da entrada original adicionando zeros nas bordas do volume de entrada, com isso, o tamanho do volume de saída se mantém igual ao de entrada após a convolução, os detalhes das bordas que então seriam perdidos são preservados para as próximas convoluções (KARN, 2016).

Figura 4. Ilustração de como o passo influencia o deslocamento de um filtro 3x3 em duas etapas sucessivas da convolução. As imagens em a) e b) correspondem a um passo unitário, enquanto as imagens em c) e d) a um passo igual a 2.



Fonte: (ARAÚJO *et al.,* 2017).

Com isso, é possível computar a altura (*AC*) e a largura (*LC*) do volume resultante de uma camada convolucional utilizando a Equação (2) e a Equação (3), respectivamente (HAYKIN, 2009).

$$\mathrm{AC} = \frac{\mathrm{A}-\mathrm{F}+2\mathrm{P}}{\mathrm{S}} + 1, \qquad (2)$$

$$\mathrm{LC} = \frac{\mathrm{L}-\mathrm{F}+2\mathrm{P}}{\mathrm{S}} + 1, \qquad (3)$$

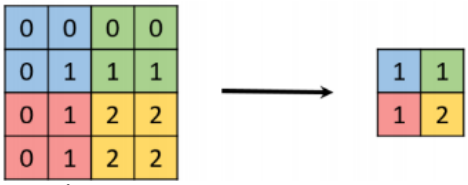
em que *A* e *L* correspondem, respectivamente, a altura e largura do volume de entrada, *F* ao tamanho dos filtros utilizados, *S* ao valor do passo, e *P* ao valor do zero-*padding*.

2.2. Camada de *Pooling*

Logo após uma camada convolucional, normalmente existe uma camada de *pooling* com o objetivo de reduzir progressivamente a dimensão espacial do volume de entrada, consequentemente a redução diminui o custo computacional da rede e evita *over-fitting* (SRIVASTAVA et al., 2014). Na operação de *pooling*, os valores que pertencem a uma determinada região do mapa de características, gerados pelas camadas convolucionais, são substituídos por alguma métrica dessa região, como por exemplo: valor máximo (*max-polling*), valor mínimo (*min-polling*) ou valor médio (*avg-polling*). A operação de *pooling* mais comum consiste em substituir os valores de uma região

pelo valor máximo (*max-polling*) (ARAÚJO et al., 2017), como ilustrado na Figura 5. Quando se utiliza essa operação, os valores resultantes das sub-regiões sobrepostas pelo filtro são definidos pelo maior valor daquela sub-região (*max*), e é útil para eliminar valores desprezíveis, reduzindo a dimensão da representação dos dados e acelerando a computação necessária para as próximas camadas, além de criar uma invariância a pequenas mudanças e distorções locais (ARAÚJO *et al.,* 2017).

Figura 5. Aplicação de *max-pooling* em uma imagem 4x4 utilizando um filtro 2x2. Além de reduzir o tamanho da imagem, consequentemente reduzindo o processamento para as próximas camadas, essa técnica também auxilia no tratamento de invariâncias locais.



Fonte: (ARAÚJO *et al.,* 2017).

A altura (*AP*) e a largura (*LP*) do volume resultante após a operação de *pooling* podem ser calculadas com a Equação (4) e Equação (5), respectivamente.

$$\mathrm{AP} = \frac{\mathrm{A}-\mathrm{F}}{\mathrm{S}} + 1, \qquad (4)$$

$$\mathrm{LP} = \frac{\mathrm{L}-\mathrm{F}}{\mathrm{S}} + 1, \qquad (5)$$

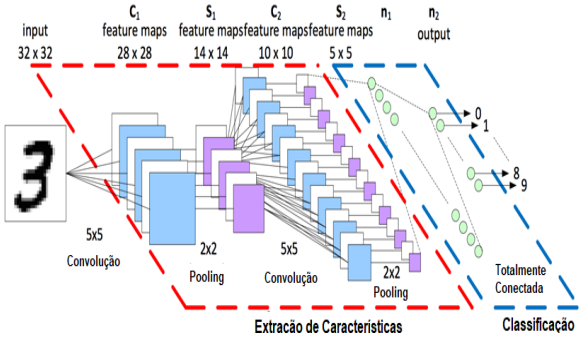
onde *A* e *L* correspondem respectivamente a altura e largura do volume de entrada, enquanto que *F* representa o tamanho da janela utilizada, e *S* é o valor do passo. Vale destacar que a profundidade do volume de entrada não é alterada pela operação de *pooling*.

2.3. Camada Totalmente Conectada

A saída das camadas convolucionais e de *pooling* representam as características extraídas da imagem de entrada. O objetivo das camadas totalmente conectadas é utilizar essas características para classificar a imagem em uma classe pré-determinada, como ilustrado na Figura 6. As camadas totalmente conectadas são exatamente iguais a uma rede neural artificial convencional (*Multi Layer Perceptron* - MLP) (ARAÚJO et al., 2017) que usa a função de ativação *softmax* (KARN, 2016)(WANG *et al.,* 2018) na última camada (de saída).

Essas camadas são formadas por unidades de processamento conhecidas como neurônio, e o termo "totalmente conectado" significa que todos os neurônios da camada anterior estão conectados a todos os neurônios da camada seguinte.

Figura 6. Ilustração da extração de características de uma imagem por uma CNN e sua posterior classificação.



Fonte: (ARAÚJO *et al.,* 2017).

Em termos matemáticos, um neurônio pode ser descrito por:

$$\mathrm{u_k} = \sum_{\mathrm{j}=1}^{\mathrm{m}} \mathrm{w_{kj}} \mathrm{x_j}, \qquad (6)$$

$$\mathrm{y_k} = \varphi(\mathrm{u_k} + \mathrm{b_k}), \qquad (7)$$

em que x_1, x_2, ..., x_m são os *m* sinais de entrada, w_{k1}, w_{k2}, e w_{km}, são os pesos sinápticos do neurônio *k*, e k_b corresponde ao viés ou bias, responsável por realizar o deslocamento da função de ativação definida por φ.

Conforme já mencionado, a última camada da rede utiliza *softmax* como função de ativação. Essa função recebe um vetor de valores como entrada e produz a distribuição probabilística da imagem de entrada pertencer a cada uma das classes na qual a rede foi treinada. Vale destacar que a soma de todas as probabilidades é igual a 1.

A técnica conhecida como *dropout* Srivastava *et al.,* (2014) também é bastante utilizada entre as camadas totalmente conectadas para reduzir o tempo de treinamento e evitar *overfitting*. Essa técnica consiste em remover, aleatoriamente a cada iteração de treinamento, uma determinada porcentagem dos neurônios de uma camada, e adicioná-los na iteração seguinte. Essa técnica também confere à rede a habilidade de aprender atributos mais

robustos, uma vez que um neurônio não pode depender da presença específica de outros neurônios.

2.4. Treinando a Rede com *backpropagation*

Inicialmente, todos os valores dos filtros das camadas convolucionais e os pesos das camadas totalmente conectadas são inicializados de forma aleatória. Em seguida, os valores são ajustados de forma a otimizar a acurácia da classificação quando considera a base de imagens utilizada no processo de treinamento.

A forma mais comum de treinamento de uma CNN é por meio do algoritmo *backpropagation* (HAYKIN, 2009). O algoritmo *backpropagation* é utilizado para calcular os valores dos gradientes do erro. Em seguida, a técnica do gradiente descendente é utilizada para ajustar os valores dos filtros e pesos na proporção que eles contribuíram para o erro total (HAYKIN, 2009). Devido ao ajuste realizado, o erro obtido pela rede é menor a cada vez que uma mesma imagem passa pela rede. Essa redução no erro significa que a rede está aprendendo a classificar corretamente imagens devido ao ajuste nos valores dos filtros e pesos. Em geral, os parâmetros número e tamanho dos filtros nas camadas convolucionais e quantidade de camadas não sofrem alterações durante o processo de treinamento.

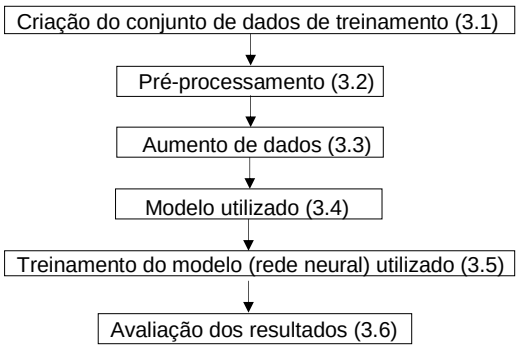
2.5. Transferência de Aprendizado

Na prática, não é frequente treinar uma CNN com inicializações aleatórias de pesos, pelo fato de que na maioria dos casos é relativamente raro ter um conjunto de dados do tamanho necessário, além disso, seriam necessárias algumas semanas de treinamento utilizando múltiplas GPUs. Com isso, uma prática comum consiste em utilizar os pesos de uma rede já treinada para uma base muito grande, como a ImageNet (IMAGENET, 2018) que possui mais de 1 milhão de imagens e 1000 classes. O primeiro passo antes de utilizar esta técnica é checar a similaridade entre o novo conjunto de dados com o conjunto de dados original (ImageNet). Constatada a similaridade entre os conjuntos, é possível utilizar os pesos para inicializar e retreinar a rede, ou mesmo para a extração de características de imagens. Uma observação importante é que as imagens devem possuir o mesmo tamanho que as imagens que foram utilizadas no conjunto de dados original (ImageNet), caso não seja, é necessário adicionar uma etapa de pré-processamento para redimensionar os dados de entrada para o tamanho necessário (KARPATHY, 2019).

3. MÉTODO PROPOSTO

Nesta seção é descrito o funcionamento do método proposto, sendo dividido em seis etapas: criação do conjunto de dados de treinamento, pré-processamento, aumento de dados, modelo utilizado, treinamento do modelo e avaliação dos resultados (Figura 7).

Figura 7. Fluxograma representando as etapas do método proposto.



Fonte: Autor (2019).

3.1. Conjunto de Dados de Treinamento

O conjunto de dados de treinamento utilizado neste trabalho é constituído por imagens das cédulas do Real, que foram obtidas a partir de uma câmera de 12 megapixels de um smartphone Galaxy S7. Essas imagens foram capturadas em diferentes pontos de vista em diversos lugares, para que se tenha uma grande variação da cena de fundo em relação às notas e diferentes condições de iluminação. Alguns exemplos de imagens capturadas são mostrados na Figura 8. No total foram obtidos duas mil e quatrocentos e cinquenta e cinco fotos de cédulas do Real dos respectivos valores de 2, 5, 10, 20, 50 e 100 reais, de ambos os lados das cédulas.

Além das imagens das cédulas obtidas com o uso da câmera do smartphone, foram utilizadas imagens do conjunto de dados ImageNet (IMAGENET, 2018). Este conjunto possui imagens variadas e que não possuem cédulas do Real, como amostrado na Figura 9. Uma classe de fundo foi adicionada ao conjunto de dados de treinamento, com a finalidade da rede neural conseguir distinguir entre características presentes nas cédulas do Real das características presentes no fundo das imagens.

Figura 8. Amostra das imagens da base de dados de treinamento.

Fonte: Autor (2019).

As imagens obtidas com o uso da câmera do smartphone possuem fundos variados, com detalhes que não devem ser considerados como partes das cédulas. Essas características de fundo devem ser desconsideradas das características das cédulas no processo de extração pela rede neural convolucional. Esse processo foi fundamental para separar as imagens que são de cédulas das imagens de fundo (HUANG, 2016).

Figura 9. Amostra de imagens de fundo da base de dados ImageNet.

Fonte: (IMAGENET, 2018).

3.2. Pré-processamento

As imagens obtidas com o uso da câmera do smartphone possuem uma resolução de 2160x2160 pixels. Foi necessário realizar uma redução na resolução das imagens do conjunto de treinamento para reduzir o custo computacional necessário para realizar o treinamento da rede neural. As imagens foram redimensionadas para a resolução de 480x480 pixels.

Após o redimensionamento das imagens do conjunto de treinamento, as imagens foram separadas em pastas nomeadas com base nos valores das cédulas do Real. Além do respectivo nome do valor (dois, cinco, dez, vinte, cinquenta e cem), é possível observar a existência de uma descrição: “Frente”, que representa o lado onde se localiza a Efígie Simbólica da República e “Verso” que representa o lado onde se encontra a gravura do animal.

3.3. Aumento de Dados

Os dados precisam ter boa diversidade, pois o objeto de interesse precisa estar presente em tamanhos, ângulos, condições de iluminação e perspectivas variadas, para que seja possível realizar o reconhecimento em diversas condições e ambientes do mundo real (PRASAD, 2017). O aumento de dados é uma técnica muito utilizada quando se trata de redes neurais convolucionais. Esse tipo de rede neural utiliza uma grande quantidade de imagens para treinamento. Por este motivo, neste trabalho foram utilizadas técnicas de aumento de dados, pois, o conjunto de imagens capturadas inicialmente possui poucas imagens. A Tabela 1 apresenta, para cada cédula do Real, as quantidades de imagens capturadas e as quantidades de imagens após o aumento de dados.

Tabela 1. Quantidade de imagens obtidas para os valores das cédulas de ambos os lados.

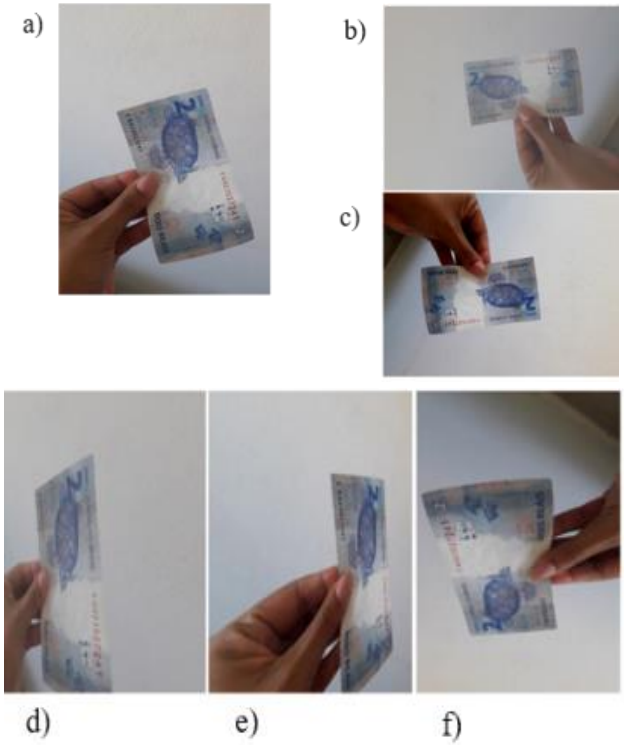
Classe / Rótulo	Quantidade de imagens capturadas	Quantidade de imagens depois do aumento de dados
Dois Verso	294	1495
Dois Frente	279	1444
Cinco Verso	155	1220
Cinco Frente	236	1369
Dez Verso	135	1151
Dez Frente	153	1200
Vinte Verso	165	1222
Vinte Frente	186	1286
Cinquenta Verso	300	1493
Cinquenta Frente	315	1528
Cem Verso	118	1118
Cem Frente	119	1128

Fonte: Autor (2019).

As operações utilizadas no aumento de dados neste trabalho foram: escala, rotação, translação, adição de ruídos (sal e pimenta),

alteração de brilho, contraste, alteração de perspectiva, inversão horizontal e inversão vertical (PRASAD, 2017). Um exemplo do resultado das operações de aumento de dados aplicadas sobre uma imagem é mostrado na Figura 10.

Figura 10. a) representa a imagem original, b) resultado de uma operação de rotação e diminuição de contraste, c) resultado de uma operação de rotação e aumento de contraste, d) resultado de uma operação de alteração de perspectiva e diminuição de contraste, e) resultado de uma operação de alteração de perspectiva e f) resultado de uma operação de rotação.



Fonte: Autor (2019).

3.4. Modelo utilizado

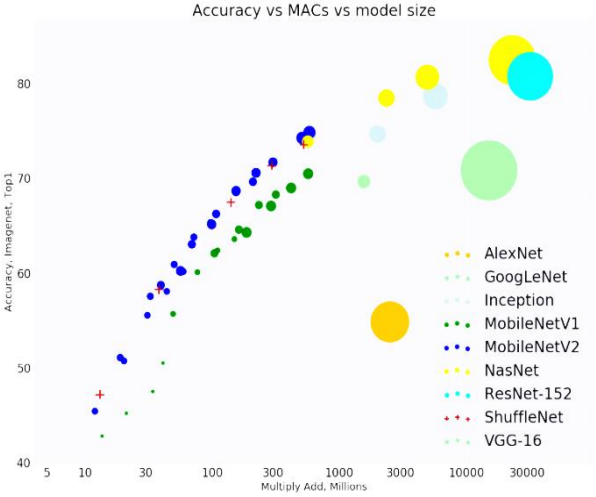
Existem diversos modelos de redes neurais convolucionais, esses modelos possuem como principal objetivo a classificação, detecção ou segmentação de objetos, a grande diferença entre os modelos são: o número de camadas convolucionais presentes, número de neurônios ocultos utilizados nas camadas totalmente conectadas e o tamanho dos filtros utilizados nas camadas convolucionais, sendo que esses parâmetros afetam diretamente na precisão e custo computacional do modelo. O modelo conhecido como MobileNetV2 (SANDLER et al., 2018) foi projetado para ser utilizado em aparelhos smartphones, possui uma boa acurácia e um baixo custo computacional. Por esse motivo esse foi o modelo de rede neural convolucional escolhido para ser utilizando neste trabalho. Na Figura 11 é mostrado um gráfico representando a comparação entre acurácia e custo computacional dos modelos de redes neurais convolucionais mais utilizados.

O MobileNetV2 é nove vezes mais eficiente do que as outras redes neurais (MobileNetV1, NasNet, ShuffleNet) comparadas em precisão (HOLLEMANS, 2018), como pode ser observado na Figura 11. Uma das vantagens de utilizar esse modelo é que ele foi treinado no conjunto de dados ImageNet (IMAGENET, 2018) que possui mais de um milhão de imagens classificadas em mil classes distintas, essa técnica de reutilização de redes neurais já treinadas é conhecida como transferência de aprendizado (KARPATHY, 2015) (Seção 2.5).

3.5. Treinamento da rede neural

Para realizar o treinamento da rede neural, foi utilizada a biblioteca TensorFlow (TENSORFLOW, 2018), essa biblioteca foi desenvolvida pelo Google Brain Team com o intuito de facilitar o desenvolvimento de aplicações que utilizam computação numérica de alto desempenho (CPUs, GPUs). Ela oferece um vasto suporte para aprendizado de máquina e aprendizado profundo (ABADI, 2016).

Figura 11. Gráfico representando a comparação nos quesitos acurácia e custo computacional entre os modelos de redes neurais convolucionais mais utilizados.



Fonte: (SANDLER et al., 2018).

O processo de treinamento foi realizado utilizando um computador com processador AMD

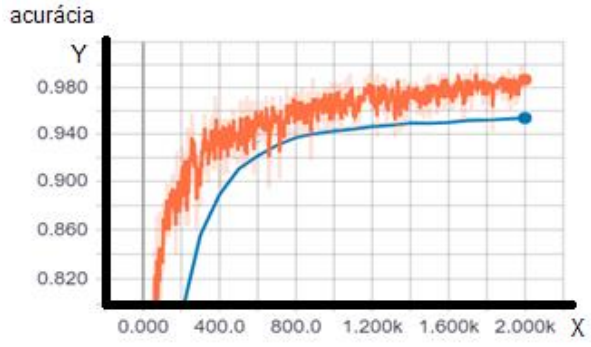
FX-4300 de 3.8GHz com 8GB de memória RAM e placa de vídeo GTX 1060 OC com 6GB.

As imagens do conjunto de dados de treinamento foram utilizadas como parâmetro de entrada para o modelo MobileNetV2, além disso, foram definidos alguns parâmetros necessários para a realização do treinamento. Esses parâmetros são conhecidos como hiper parâmetros, eles são fixos e não mudam ao decorrer do treinamento, são fundamentais para a eficiência da rede neural. Os hiper parâmetros utilizados no treinamento foram: taxa de aprendizado: 0,045; tamanho do lote: 256; número de épocas: 2000; porcentagem de imagens para validação: 30% das imagens do conjunto de treinamento. Todos estes parâmetros foram refinados empiricamente com base nos testes realizados.

O tempo de treinamento foi de uma hora e oito segundos com base nas duas mil épocas que foram definidas. Em cada passo foram processados duzentas e cinquenta e seis imagens do conjunto de treinamento por vez.

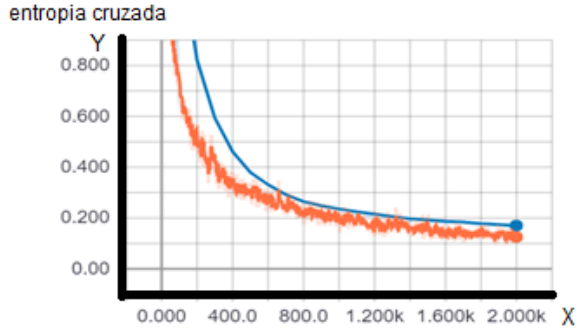
O resultado final do processo de treinamento atingiu uma acurácia de 97.4% baseada nos 30% das imagens definidas para validação. É possível observar o progresso do treinamento visualizando os gráficos gerados pelo Tensorboard que é um conjunto de ferramentas do TensorFlow para simplificar a visualização dos dados gerados durante o treinamento. Na Figura 12 é mostrado o progresso em relação à acurácia, e na Figura 13 é mostrado o erro gerado pela rede durante as duas mil épocas.

Figura 12. Gráfico gerado pelo Tensorboard que representa a acurácia do aprendizado durante as duas mil épocas. O eixo Y representa a acurácia e o eixo X representa as épocas. A linha de cor laranja está relacionada à acurácia do treino, e a linha azul está relacionada à acurácia da validação.



Fonte: Autor (2019).

Figura 13. Gráfico gerado pelo Tensorboard que representa o erro durante as duas mil épocas. O eixo Y representa a taxa de erro e o eixo X representa as épocas. A linha de cor laranja está relacionada ao erro do treino, e azul ao erro da validação.



Fonte: Autor (2019).

3.6. Avaliação dos Resultados

Para realizar a avaliação final da rede neural, foram utilizadas 60 imagens para validação, 10 imagens para cada valor da nota do Real. Essas imagens não foram incluídas no conjunto de dados de treinamento.

O método proposto atingiu resultados satisfatórios no processo de reconhecimento das cédulas do Real, dentro do total das 60 imagens do conjunto utilizadas para validação, 55 imagens foram classificadas corretamente e 5 incorretamente, como pode ser visualizado na Tabela 2. Foi obtida uma acurácia final de 91,67%, e o tempo médio do processamento de cada imagem foi de aproximadamente 200 milissegundos. Com isso, foi possível notar que a abordagem utilizada é viável e eficiente para ser utilizada em cenários reais onde se tem um ambiente com alguma variação de iluminação, fundos variados e capturas de imagens em diferentes pontos de vista.

Tabela 2. Resultados obtidos na validação da Rede Neural.

Cédula	Quantidade	Acerto	Erro	Acertos [%]
Dois	10	10	0	100
Cinco	10	9	1	90
Dez	10	9	1	90
Vinte	10	10	0	100
Cinquenta	10	9	1	90
Cem	10	8	2	80

Fonte: Autor (2019).

A matriz de confusão apresentada na Tabela 3 ilustra a eficácia do método proposto

neste trabalho. É possível observar que a maior parte das imagens contendo notas do Real foi classificada corretamente, exceto uma nota de Cinco que foi classificada como Dez, uma nota de Dez que foi classificada como Cinco, uma nota Cinquenta que foi classificada como Dez, e duas notas de Cem que foram classificadas, uma como Dez e outra como Cinquenta.

Tabela 3. Matriz de confusão obtida com a validação da Rede Neural.

	Dois	Cinco	Dez	Vinte	Cinquenta	Cem
Dois	10	0	0	0	0	0
Cinco	0	9	1	0	0	0
Dez	0	1	9	0	0	0
Vinte	0	0	0	10	0	0
Cinquenta	0	0	1	0	9	0
Cem	0	0	1	0	1	8

Fonte: Autor (2019).

4. CONSIDERAÇÕES FINAIS

Os resultados deste trabalho mostram que a metodologia desenvolvida pode ser aplicada em cenários que exigem uma resposta rápida e com acurácia. O tempo de processamento para cada imagem foi em torno de 200 milissegundos, com uma acurácia final 91,67% no reconhecimento de cédulas do Real. Foi possível notar que a utilização da rede neural convolucional foi fundamental no reconhecimento de cédulas do Real, apresentando um bom comportamento mesmo com imagens capturadas em cenários não controlados, o que é muito difícil de se alcançar com uso de outras técnicas como as aplicadas por Mombach (2015), que também reconhecem cédulas do Real. Nesse trabalho os autores conseguiram uma acuraria de 89% com tempo de processamento baixo, porém, utilizando um ambiente controlado e com o uso de técnicas de correspondência de pontos chave e identificação de cores das cédulas.

Como trabalhos futuros, acredita-se que melhorias podem ser realizadas no conjunto de dados de treinamento, adicionando novas imagens representando cenários que o conjunto de dados de treinamento utilizado não abordou. Além da melhoria no conjunto de dados de treinamento, é possível refinar os parâmetros (taxa de aprendizado, tamanho do lote, porcentagem de imagens de validação) que foram definidos no treinamento, buscando encontrar os melhores parâmetros que quando aplicados podem proporcionar uma melhor acurácia no reconhecimento.

O modelo de rede neural convolucional MobileNetV2, que foi utilizado neste trabalho, é muito propício para ser utilizado em um smartphone, devido às suas características de boa acurácia e baixo custo computacional. Propõe-se para um trabalho futuro, utilizando a metodologia desenvolvida neste trabalho, a construção de um aplicativo para o reconhecimento das notas do Real utilizando os frames capturados diretamente pela câmera do dispositivo. Esse aplicativo poderia ser utilizado no dia a dia por deficientes visuais auxiliando-os na identificação das cédulas monetárias do Real.

REFERÊNCIAS

ABADI, M. Tensorflow: Large-scale Machine Learning on Hererogeneous Distributed Systems. **Distributed, Parallel and Cluster Computing**, p. 1-19, 2016.

AGARAP, A. F. Deep Learning using Rectified Linear Units (ReLU). **Neural and Evolutionary Computing**, v. 1, 2018.

ARAÚJO, F. H. D.; CARNEIRO, A. C.; SILVA, R. V.; MEDEIROS, F. N. S.; USHIZIMA, D. M. Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. *In*: ESCOLA REGIONAL DE INFORMÁTICA DO PIAUÍ. 3., **Anais[...].** ERI 2017, v. 1, n. 1, Piauí, p. 382-406, 2017.

FUNDAÇÃO DORINA NOWILL. **Estatísticas da deficiência visual.** 2010. Disponível em: https://www.fundacaodorina. org.br/a-fundacao/deficiencia-visual/estatisticas-da-deficiencia-visual. Acessado em: 10 abr. 2019.

GHELLERE, J. S. **Detecção de objetos em imagens por meio da combinação de descritores locais e classificadores**. 2015. Trabalho de Graduação, Departamento de Computação, Universidade Tecnológica Federal do Paraná, Medianeira, Brasil, 2015.

GUO, T.; DONG, J.; LI, H.; GAO, Y. Simple convolutional neural network on image classification. *In*: IEEE INTERNATIONAL CONFERENCE ON BIG DATA ANALYSIS (ICBDA). 2., Beijing, China, 2017. https://doi.org/10.1109/ICBDA.2017.8078730

HAYKIN, S. **Neural Networks and Learning Machines**. McMaster University, Ontario Canada, 3rd ed., 2009.

HOLLEMANS, M. **MobileNet version 2**, 2018. Disponível em: http://machinethink.net/blog/mobilenet-v2. Acesso em: 11 nov. 2018.

HUANG, J. **Tensorflow detection model zoo**. 2016. Disponível em: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md. Acessado em: 12 jan. 2019.

IDE, H.; KURITA, T. Improvement of learning for CNN with ReLU activation by sparse regularization. *In*: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), Anchorage, AK, USA, 2017. https://doi.org/10.1109/IJCNN.2017.7966185

ImageNet. 2018. Disponível em: http://www.image-net.org. Acesso em: 10 jan. 2019.

KARN, U. An Intuitive Explanation of Convolutional Neural Networks, 2016. Disponível em: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets. Acesso em: 17 nov. 2018.

KARPATHY, A. **Transfer learning and fine-tuning convolutional neural networks**. 2015. Disponível em: http://cs231n.github.io/transfer-learning. Acesso em: 14 set. 2018.

KARPATHY, A. **CS231n Convolutional Neural Networks for Visual Recognition**. Disponível em: http://cs231n.github.io/convolutional-networks. Acesso em: 15 jan. 2019.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural Computation**, v.1, n. 4, p. 541-551, 1989.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. v. 86, n. 11, **Proceedings** of the IEEE, p. 2278-2324, 1998. https://doi.org/10.1162/neco.1989.1.4.541

LECUN, Y. Lenet-5, convolutional neural networks. 2015. Disponível em: http://yann.lecun.com/exdb/lenet. Acessado em: 06 set. 2018. https://doi.org/10.1109/5.726791

MOMBACH, J.G. **Proposta de aplicativo móvel para identificação de cédulas de real por pessoas com deficiência visual**. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Pampa, Alegrete, Brasil, 2015.

OLIVEIRA, H. S. **Redes Neurais Convolucionais para Classificação de Expressões Faciais de Emoções**. Trabalho de Conclusão de Curso (Graduação) - Universidade Tecnológica Federal de Roraima, Medianeira, 2017.

PRASAD, P. **Data Augmentation Techniques in CNN using Tensorflow.** 2017. Disponível em: https://medium.com/ymedialabs-innovation/data-augmentation-techniques-in-cnn-using-tensorflow-371ae43d5be9. Acesso em: 12 jan. 2019.

. COMO os cegos diferenciam as notas de dinheiro? **Revista Época** 2009. Disponível em: http://revistaepoca.globo.com/Revista/Epoca/0,,EMI103120-15223,00-COMO+OS+CEGOS+DIFERENCIAM+AS+NOTAS+DE+DINHEIRO.html. Acesso em: 08 nov. 2018.

SÁ, E. D. **Informática para as pessoas cegas e com baixa** visão, 2006. Disponível em: http://www.bancodeescola.com/info_para_cegos.htm. Acesso em: 20 maio 2018.

SANDLER, M.; HOWARD, A.G.; ZHU, M.; ZHMOGINOV, A.; CHEN, L. MobileNetV2: Inverted Residuals and Linear Bottlenecks. **Computer Vision and Pattern Recognition**, 2018. https://doi.org/10.1109/CVPR.2018.00474

SILVA, C.; WELFER, D.; GIODA, F. P.; DORNELLES, C. Cattle Brand Recognition using Convolutional Neural Network and Support Vector Machines. **IEEE Latin America Transactions**, v. 15, n. 2, p. 310-316, 2017. https://doi.org/10.1109/TLA.2017.7854627

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, n. 15, p. 1929-1958, 2014.

TensorFlow. 2018. Disponível em: https://www.tensorflow.org. Acesso em: 20 out. 2018.

VEQUETINE, V.; ZANCHETTA, M.; BRAGA, J. **Método para auxiliar o reconhecimento de cédulas monetárias pelos deficientes visuais**. 2013. Faculdade de Computação (FACOM). Disponível em: www.lbd.dcc.ufmg.br/colecoes/wim/2013/0024.pdf. Acesso em: 18 jan. 2019.

WANG, F.; CHENG, J.; LIU, W.; LIU, H. Additive margin softmax for face verification. **IEEE Signal Processing Letters**, v. 25, p. 926-930, 2018. https://doi.org/10.1109/LSP.2018.2822810

WU, J. Introduction to Convolutional Neural Networks. 2017. National [Online]. Key Lab for Novel Software Technology, Nanjing University, China. Disponível em: https://cs.nju.edu.cn/wujx/paper/CNN.pdf. Acesso em: 13 jan. 2019.

YANG, J.; LI, J. Application of deep convolution neural network. *In*: IEEE INTERNATIONAL COMPUTER CONFERENCE ON WAVELET ACTIVE MEDIA TECHNOLOGY AND INFORMATION PROCESSING (ICCWAMTIP), 14., 2017. Anais […]. Chengdu, China, 2017. https://doi.org/10.1109/ICCWAMTIP.2017.8301485