Reconhecimento de Display de 7-Segmentos com Redes Neurais

Guilherme Araujo Santos¹\*;Diego Pedroso dos Santos2

1 Instituto de Pesquisas Tecnológicas. Técnico no laboratório de Anemometria. Avenida Professor Almeida Prado,532 – Butantã; 05508-901 São Paulo, SP, Brasil

2 Mestre em Engenharia de Controle e Automação. Avenida Marques de São Vicente, 2898 – Água Branca; 05034-040 São Paulo, São Paulo, Brasil

\*autor correspondente: glrme\_araujo@hotmail.com

Reconhecimento de Display de 7-Segmentos com Redes Neurais

Resumo

Este projeto foi desenvolvido com o objetivo de aprimorar e automatizar a aquisição de dados localizados em displays de 7 segmentos, fotografando o display do equipamento e processando área que contém os dígitos, para então ser reconhecido digitalmente qual o digito na foto, utilizando uma Rede Neural Convolucional [CNN] que se mostrou satisfatória na tarefa de reconhecer esses dígitos a partir dessas imagens. E a partir desse conjunto, essa ferramenta foi implementada na calibração de equipamentos de medição no laboratório de anemometria localizado no Instituto de Pesquisas Tecnológicas (IPT). Durante o desenvolvimento, foi percebido a complexidade da construção de uma rede neural, sua aplicação prática, os excelentes resultados alcançados, e a influência da aleatoriedade durante o treinamento de um modelo de predição.

**Palavras-chave:** Rede Neural Convolucional (CNN); Redes Neurais Profundas; Visão Computacional; Processamento de Imagens; Display 7-Segmentos.

**Abstract**

This project was developed with the aim of enhancing and automating the acquisition of data from 7-segment displays. It involves capturing photographs of the equipment's display and processing the area containing the digits. Subsequently, a Convolutional Neural Network (CNN) is employed to digitally recognize the digits in the photo. The CNN has proven to be effective in this task of digit recognition from such images. This innovation has been implemented in the calibration of measurement equipment at the anemometry laboratory located in the Institute of Technological Research (IPT). Throughout the development, the complexity of constructing a neural network, its practical application, and the excellent results achieved were observed, along with the influence of randomness during the training of a prediction model.

**Keywords:** Convolutional Neural Network (CNN); Deep Neural Networks; Computer Vision; Image Processing; 7-Segment Display.

Introdução

Atualmente, os displays de 7 segmentos permanecem amplamente empregados em diversos dispositivos, incluindo painéis de controle, aparelhos domésticos e medidores digitais. O reconhecimento de dígitos se tornou uma ferramenta essencial em diversas áreas e processos, se tornando uma das mais fundamentais das áreas do processamento de imagens e visão computacional.

Uma das técnicas mais avançadas e confiáveis para esse tipo de reconhecimento é encontrada na área de Redes Neurais Profundas, inspiradas no funcionamento do cérebro humano, projetadas para realizar tarefas específicas de aprendizado e reconhecimento de padrões, resolver diversas tarefas complexas e não-lineares. A rede neural que mais apresenta uma aplicação direcionada a essa proposta, é a Rede Neural Convolucional (CNN - Convolutional Neural Network), desenvolvida por Yann LeCun em 1998 (Entendendo as Redes Neurais Convolucionais. 2018), que é indicada para o processamento eficiente de dados bidimensionais, como imagens e vídeos. Elas se destacam em tarefas relacionadas à visão computacional e são amplamente utilizadas em aplicações de reconhecimento de padrões de dados em formato de grade, como imagens. A sua arquitetura e funcionamento tem ótimo desempenho na visão computacional. As CNNs são ferramentas poderosas para o processamento de dados visuais, desempenhando um papel crucial no avanço de tecnologias relacionadas à visão computacional e reconhecimento de padrões.

O trabalho tambem destaca a importância de uma base de dados extensa e diversificada no treinamento de algoritmos de aprendizagem profunda. A base de dados (Connor Monahan. 2020) utilizada se apresentou satisfatória vários fatores que implicam na qualidade final do modelo. Além disso, é ressaltada a necessidade de um computador compatível com o processamento requerido. No contexto de redes neurais, explica a utilização de resultados conhecidos para treinar o algoritmo, a importância de ajustar os pesos entre neurônios durante as iterações, e como as épocas permitem observar a evolução do modelo. Destacando a relevância da variedade nos dados para alcançar resultados satisfatórios. Mostrando a importância da combinação adequada de dados, poder de processamento e configurações de treinamento para obter sucesso em algoritmos de aprendizagem profunda.

A solução proposta envolve o uso de uma câmera para fotografar o dígito exibido, processar a área de interesse da foto, e aplicar um modelo de rede neural convolucional para realizar a predição, contribuindo para uma automação do processo e precisão na aquisição dos dados, superando problemas de tempo e erros humanos na transcrição.

**Material e Métodos**

Esta seção apresenta informações referentes ao contexto, display, processo de calibração, coleta dos dados, processamento, recursos computacionais e modelagem do algoritmo.

**Contexto**

O trabalho além de apresentar a construção de uma rede neural para reconhecimento dos dígitos, também apresenta a implementação desse algoritmo dentro de um cenário real. Existem inúmeros processos que contam com a presença de equipamentos que utilizam um display de 7-segmentos em sua arquitetura, esses equipamentos podem ser de medição, relógios, ou indicadores, a sua utilização é algo amplo e que há muito tempo é utilizado, pois sua arquitetura e funcionamento são eficientes.

A contextualização se refere a necessidade de automatizar a aquisição dos dígitos impressos em um display de 7-segmentos, presente nos equipamentos que são usados na calibração de anemômetros (equipamentos de medição do ar), no Laboratório de Anemômetria, no Instituto de Pesquisas Tecnológicas (IPT). O executor da calibração tem como umas das etapas do processo de calibração, a tarefa de visualizar os dígitos exibidos pelo equipamento e anota-los no formulário de calibração, com o passar do tempo se mostrou um problema nos processos, pois demandava de tempo e suscetível a erros de transcrição feitas pelo usuário.

A solução proposta apresenta a implementação de uma câmera para fotografar o digito do display de 7- segmentos, exibido pelo equipamento a ser calibrado, e a partir dessa fotografia, realizar o processamento da imagem e utilizar um algoritmo de rede neural para realizar a predição, e automatizar sua aquisição.

**Display**

Um "display" é um termo que geralmente se refere a um dispositivo de saída que exibe informações visuais, como texto, gráficos, imagens ou vídeos. O principal propósito de um display é fornecer uma interface visual para o usuário interagir e consumir informações a partir de um dispositivo eletrônico. Displays são encontrados em uma variedade de dispositivos eletrônicos, incluindo: monitores, televisões, equipamentos eletrônicos (balanças, eletrodomésticos, etc.), painéis de controle.

Em 1950, houve um impulso significativo no avanço da eletrônica digital. Antes da introdução dos displays de 7 segmentos, eram empregados displays de cátodo frio e outras técnicas para representar dados numéricos. No entanto, essas abordagens eram frequentemente grandes em tamanho e demandavam uma quantidade substancial de energia. Já em 1960, com o avanço da tecnologia, os displays de cátodo frio foram sendo substituídos por displays de cátodo quente e, mais tarde, pelos displays de 7 segmentos.

Inicialmente, os displays de 7 segmentos eram usados em calculadoras e relógios digitais. Sua capacidade de representar números de forma clara e compacta fez com que se tornassem uma escolha popular para dispositivos eletrônicos que exigiam exibição numérica. Ao longo das décadas, a tecnologia por trás dos displays de 7 segmentos continuou a evoluir. Inicialmente, eram frequentemente construídos com lâmpadas ou LEDs individuais para cada segmento. Posteriormente, os displays de LED se tornaram predominantes devido à eficiência energética e durabilidade.

O display de 7 segmentos se tornou popular devido à sua simplicidade e eficiência na representação de dígitos, letras e alguns caracteres especiais por meio de sete segmentos individuais. Cada segmento pode ser ligado ou desligado de forma independente, permitindo a representação de números de 0 a 9 e algumas letras. De maneira fundamental, um display de sete segmentos é essencialmente um conjunto de 7 LEDs dispostos de forma fixa para criar um número 8. Esses LEDs podem ser ativados individualmente, permitindo a exibição de quantidades em sistemas numéricos como decimal, binário, octal ou hexadecimal. Na configuração mais comum, eles podem ser do tipo anodo comum ou catodo comum, indicando que os LEDs são acionados por meio de níveis lógicos "0" e "1", respectivamente (Prática 04 - BCD e Display 7 Segmentos).

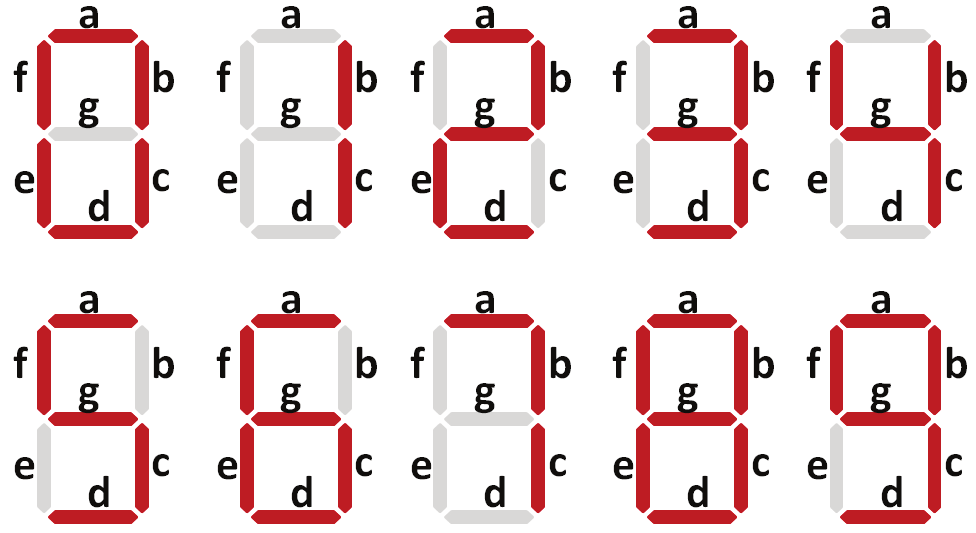


Figura 1. Exemplo de segmentos ligados para exibir dígitos de 0 a 9

Fonte: Barros (2020)

Hoje, os displays de 7 segmentos ainda são amplamente utilizados em uma variedade de dispositivos, como painéis de controle, aparelhos domésticos, medidores digitais, etc. Apesar do surgimento de tecnologias mais avançadas, os displays de 7 segmentos continuam a ser uma opção viável e econômica para muitas aplicações.

**Coleta de dados**

Uma base de dados, ou dataset, são um conjunto de informações de entradas (sinais) e saída (resultados) de fenômenos específicos que serão aprendidos pelo algoritmo escolhido. Tem o objetivo de modelar uma rede neural, ou qualquer outro algoritmo de aprendizado de máquina, gerando assim um modelo para predição de resultados.

A base de dados é utilizada foi a Sevensegdataset (Connor Monahan, 2020), ela é composta por 17279 dígitos com proporção de 28x28 pixels. A base de dados dispõe de diferentes posições, formatos, ângulos e intensidade de cor, a maior parte dos dígitos são de display de 7-segmentos, tornando ela uma base de dados variada, o que futuramente auxiliará para a modelagem de uma rede robusta.

Para cada digito, é informado seu respectivo valor. A variedade de dígitos, torna a base de dados bem robusta e confiável.

Figura 2. Exemplos dos dígitos do conjunto de dados Sevensegdataset

Fonte: Dados originais da pesquisa

A base de dados foi dividida em dois conjuntos (teste e treino), o conjunto de treino é composto por 80% dos dados (13.823 observações), que foi utilizado para treinar o algoritmo. E o conjunto de teste é composto pelos 20% restante (3.456 observações), que foi utilizado para testar a acurácia do algoritmo e validar ele.

Os dados a serem preditos são extraídos das fotografias dos equipamentos durante a sua calibração. Inicialmente é foi posicionado a câmera de forma em que o display do equipamento fique dentro do seu campo de visão sem que sofra de inclinação ou de distância longa, tudo isso influenciará a qualidade da fotografia do dígito a ser reconhecido. Então, após esse posicionamento, é selecionado manualmente a área que contém a informação a ser extraída da foto, essa área contém apenas o display do equipamento (Figura 3).



Figura 3. Selecionando a área do display na fotografia (Quadrado Azul)

Fonte: Dados originais de pesquisa

Após selecionar a área que contém o display que imprime os dígitos, foi realizada uma seleção manual de cada digito, para ser processado individualmente e não sofrer ruídos ou influência do digito próximo, sempre sendo feito o processamento na mesma área. Dessa forma, teremos uma rotina de fotografia fixa, e com os dígitos sendo separados e processados individualmente. A figura 4 ilustra a seleção de um único digito dentro da área selecionada anteriormente. Após essa seleção, acontece o processamento.

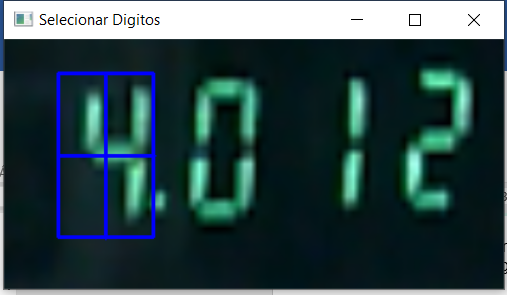


Figura 4. Selecionando um único dígito a ser processado (Quadrado Azul) Fonte: Dados originais de pesquisa

**Processo de Calibração**

O laboratório de Anemometria localizado no Instituto de Pesquisas Tecnológicas, dispõe de um túnel de vento (Figura 5), que foi construído para gerar ventos que são o essencial para o processo de calibração de equipamentos que medem a velocidade do ar.



Figura 5. Túnel de Vento

Fonte: Dados originais de pesquisa

O processo de geração do vento consiste em captar o vento por uma entrada de ar localizada na lateral traseira do túnel, após a captação o vento então é filtrado e retificado, sendo direcionado para área de contração do túnel, que reduz a secção do escoamento, fazendo com que a velocidade dele aumente, em seguida, o vento é lançado para fora do túnel, diretamente na área de teste. Durante o processo de calibração, os equipamentos a serem calibrados são posicionados nessa área de teste e seus resultados são comparados com os resultados dos equipamentos padrões do laboratório. Os equipamentos são submetidos a diversas velocidades durante o processo.

**Processamento dos dados**

O processamento das imagens segue as seguintes etapas: aplicar escala cinza, realizar o threshold, e por fim, dilatar os traços. Ao receber o digito, é aplicado uma camada de escala cinza, que tem como objetivo, deixar a foto sem cor, apenas com tons de cinza, isso facilita a aplicação de alguns tipos de técnicas de processamento de imagens, o threshold é uma dessas técnicas. A limiarização, conhecida também como threshold, é uma técnica de processamento de imagens que envolve a segregação de pixels em duas categorias distintas, geralmente preto e branco, mediante a utilização de um valor limiar. A seleção desse limiar é orientada por características específicas da imagem que se almeja realçar ou negligenciar (Python e OpenCV: Explorando Threshold e Filtros,2023).

Aplicando essa técnica, é obtido uma imagem apenas com tons de preto e branco, deixando o dígito muito nítido e com fácil identificação, além de remover ruídos ou traços indesejados. E por fim, é realizado a dilatação, que é uma operação morfológica comumente utilizada em processamento de imagens, para ampliar ou engrossar regiões específicas de uma imagem. A dilatação é frequentemente aplicada a imagens binárias, onde os pixels têm valores de 0 (preto) ou 255 (branco). Após esse processamento, é obtido a imagem final, que terá sua proporção alterada para uma proporção de 28x28 pixels, ou seja, uma matriz de 28x28. Será aplicado nessa matriz, o modelo de reconhecimento.

C:\Users\guilhermeas\digito_13.png

Figura 6. Fotografia do digito 4 processado. Fonte: Dados Originais de Pesquisa.

**Algoritmo de Reconhecimento**

O algoritmo de reconhecimento utilizado neste trabalho é a rede neural convolucional [CNN] - que será responsável por reconhecer os dígitos no projeto, é um algoritmo de aprendizado de máquina robusto e confiável para esse tipo de problema. Em 1962, Hubel e Wiesel realizaram uma pesquisa que indicou que alguns neurônios ficam ativos juntos quando expostos a linhas ou curvas, o que facilita o reconhecimento visual. Então as CNN’s realizam a filtragem de linhas, curvas e bordas, e, em cada camada subsequente, convertem essa filtragem em uma imagem mais complexa. A primeira aplicação com sucesso de uma CNN foi desenvolvida por Yann LeCun em 1998, com sete camadas entre convoluções e fully connected. As entradas de dados da CNN é uma matriz tridimensional (altura x largura x cores), em geral são utilizados 3 canais de cores, que são o vermelho, azul, e verde, o termo usado é RGB.

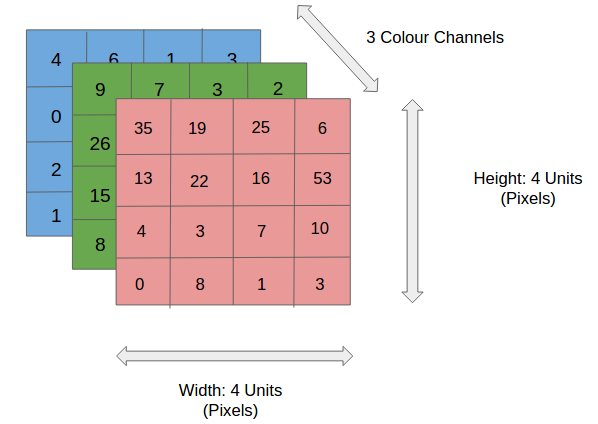


Figura 7. Exemplo de matriz tridimensional com 3 canais de cores

Fonte: Alves (2018)

A matriz então passa por uma camada de convolução, que tem o objetivo de filtrar cada pixel dos canais, e selecionar os pixels mais significativos, reduzindo o tamanho da matriz. A região da entrada onde o filtro é aplicado é chamada de “receptive field”. O filtro, que também é conhecido por kernel, é formado por pesos inicializados aleatoriamente, atualizando-os a cada nova entrada durante o processo de “backpropagation”. A profundidade da saída de uma convolução é igual a quantidade de filtros aplicados. Quanto mais profundas são as camadas das convoluções, mais detalhados são os traços identificados.

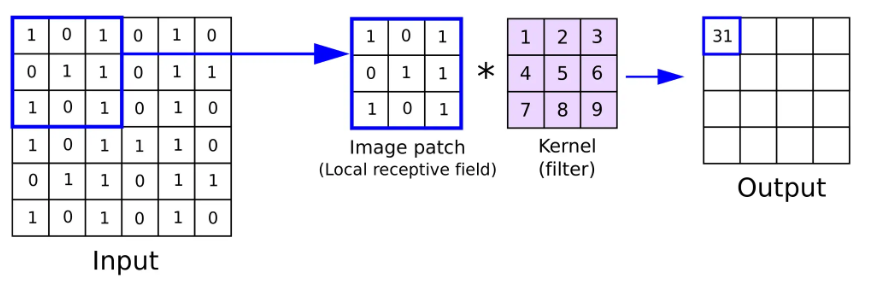


Figura 8. Exemplo da camada de convolução

Fonte: SuperAnnotate (2023)

As funções de ativação servem para trazer a não-linearidades ao sistema, para que a rede consiga aprender qualquer tipo de funcionalidade. Podem ser encontradas no final da camada. Há muitas funções, como “sigmoid”, “tanh”, “softmax” e “Relu”, foram utilizadas as funções de ativação “softmax” e “Relu”.

A “Relu” por ser mais eficiente computacionalmente sem grandes diferenças de acurácia quando comparada a outras funções, foi usada nesse desenvolvimento. Essa função zera todos os valores negativos da saída da camada anterior.

A “softmax” é uma generalização da função logística para várias dimensões e é usada na regressão logística multinomial, converte um vetor de K números reais em uma distribuição de probabilidade de K resultados possíveis.

Após a camada de convolução é encontrada a camada de “pooling”, que também tem como funcionalidade a redução da matriz de convolução anterior. O método mais comum é o “maxpooling”, no qual apenas o maior número da unidade é transmitido para a saída. Essa simplificação dos dados tem a função de reduzir a quantidade de pesos a serem aprendidos e também ajuda a evitar o “overfitting”. O usuário define o tamanho da região a ser reduzida, o mais comum é usar uma região de 2x2.



Figura 9. Exemplo da camada de Pooling (Maxpooling)

Fonte: Paperwithcode (2017)

A arquitetura da rede, convolução, função de ativação, “pooling”, pode ser feita diversas vezes dentro de uma rede neural, até chegar nos estágios finais, “flaten”, “fully connected” e função de ativação. O “Flatten” é uma camada da rede neural que é encontrada após o usuário realizar o último “pooling”. O objeto do “flaten” é de achatar a matriz inteira em uma dimensão, se tornando uma espécie de “fila” de dados, facilitando o processamento posterior.

Ao final da rede é colocada uma camada “Fully connected”, onde sua entrada é a saída da camada anterior (“flatten”) e sua saída são N neurônios, com N sendo a quantidade de classes do seu algoritmo para finalizar a classificação.

Na maioria dos casos, esta camada conta com uma função de ativação, foi utilizada a “softmax” no trabalho. A qual nos fornece a probabilidade da matriz de entrada (digito do display) pertencer a determinada classe (digito real). Assim é a estrutura básica de uma rede neural convolucional, existem outros tipos de camadas com suas aplicações e efeitos diferentes, mas os componentes principais são esses apresentados. A figura 9 ilustra a construção completa de uma CNN básica.

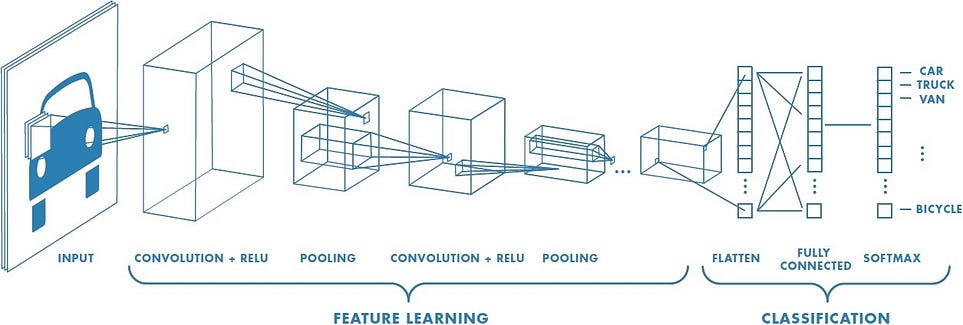


Figura 10. Exemplo de uma CNN completa

Fonte: SuperAnnotate (2023)

**Recursos Computacionais**

Para desenvolver o trabalho, foi necessário o uso de alguma linguagem de programação capaz de realizar a modelagem de uma rede neural, processamento de imagem, e controle da câmera. Devido a extensa referência, bibliotecas e poder computacional, a versão 3.10 do Python foi a linguagem a ser utilizada, sendo necessário utilizar das ferramentas disponibilizadas pelas diversas bibliotecas utilizadas no projeto, as principais foram; Keras, Tensorflow, OpenCV, todas de código aberto

Quando um software é “código aberto”, isso significa que seu código-fonte está disponível abertamente, com uma licença de direito autoral que permite visualizar/estudar, modificar e também distribuir o software de graça para qualquer pessoa ou empresa, para qualquer finalidade.

TensorFlow e Keras estão relacionados no contexto do aprendizado de máquina e das redes neurais, ambos juntos proporcionam uma poderosa ferramenta para desenvolvimento de modelos de aprendizado de máquina. O TensorFlow se beneficia da simplicidade do Keras, enquanto ainda fornece toda a flexibilidade e recursos de uma biblioteca de aprendizado de máquina de baixo nível, porem ambos necessitam de uma máquina que tenha um hardware compatível com as especificações descritas em suas documentações. Sem um computador com esse poder de processamento, não seria possível modelar a rede neural e treiná-la.

Em 2011, a equipe do Google Brain desenvolveu o sistema DistBelief, focado em redes neurais e deep learning. O sucesso do DistBelief levou várias divisões do Google a adotá-lo em aplicações comerciais e pesquisas. Diante do crescente uso interno, o Google decidiu aprimorar o sistema, visando torná-lo mais simples e rápido. Esse esforço resultou na transformação do DistBelief no TensorFlow. Atualmente em sua segunda geração (2.0.0), foi lançado como código aberto em novembro de 2015, com a versão 1.0.0 disponibilizada em fevereiro de 2017. Em janeiro de 2018, o Google anunciou o lançamento do TensorFlow 2.0, representando uma evolução significativa do framework de código aberto.

O Keras é uma ferramenta que simplifica a construção e treinamento de redes neurais, além de disponibilizar uma variedade de redes pré-treinadas para diversas tarefas. Desenvolvido com foco em facilitar a experimentação rápida, é especialmente recomendado para iniciantes no aprendizado de redes neurais devido à sua simplicidade de programação.

OpenCV (Open Source Computer Vision Library) é uma biblioteca de software de código aberto amplamente reconhecida na área de visão computacional e aprendizado de máquina. Essa biblioteca é extensivamente empregada em diversas aplicações, tais como processamento de imagem e vídeo, detecção de objetos, reconhecimento facial e uma ampla gama de outras tarefas relacionadas à visão por computador. Sua popularidade advém da sua versatilidade e eficácia na implementação de algoritmos para análise de dados visuais em tempo real (Access Camera and Display Image Using Python and Opencv. 2023).

**Resultados e Discussão**

**Modelagem do Algoritmo**

Neste trabalho, a rede foi construída desde o início, selecionando e ajustando todas as camadas necessárias paras desenvolver uma rede eficiente. Começando pela camada de entrada, ajustada para receber uma matriz de 28x28 pixels, com apenas 1 canal de cor, já que os dígitos oscilam entre preto e branco.

A primeira camada é a de convolução, com proporção de 32x32, com um kernel que irá ler uma região de 3x3 pixels até completar a matriz de 32x32.

Após essa convolução, será aplicado uma camada de pooling de 2x2 que irá selecionar os maiores valores na matriz, dessa forma irá reduzir a proporção da rede e diminuir os pesos, evitando um overfitting.

A terceira camada, também uma cada de convolução, porém ao invés de utilizar uma camada de 32x32, é utilizada uma de 64x64, ampliando a área para um detalhamento maior e reconstrução otimizada. E na quarta camada, é realizado um pooling exatamente igual ao da segunda camada.

A quinta camada, é marcada pela presença do Flatten, que “achata a matriz” transformando-a em uma fileira de pixels, o que facilita o processamento posterior.

A sexta e sétima camada são responsáveis por classificar em 100 classes diferentes, e depois para 10 classes diferentes (10 dígitos), acompanhada de uma função de ativação “softmax”, que calcula e imprime a probabilidade daquela matriz pertencer a cada uma das classes.

A figura 11 possibilita-nos ver as mudanças na estrutura e nos parâmetros no desenvolvimento da rede.

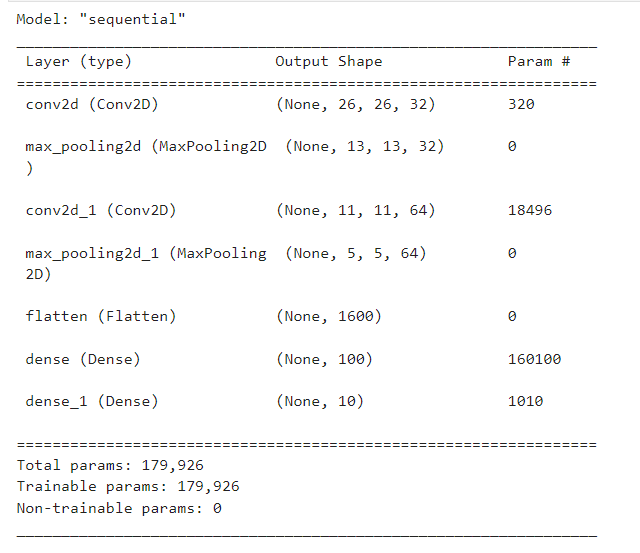


Figura 11. Arquitetura da CNN

Fonte: Resultados originais da pesquisa

**Treinamento e Validação do Algoritmo**

Para treinar um algoritmo de aprendizagem profunda, independentemente do tipo de técnica a ser aplicada em um projeto, é necessário ter uma extensa base de dados e um computador que seja compatível com o processamento. O treinamento de um algoritmo consiste em utilizar resultados conhecidos oriundos de determinados casos na resolução de outro caso com algum grau de semelhança entre eles.

Para obter bons resultados ao treinar um algoritmo, é preciso que sua base de dados, além de extensa, tenha uma boa variedade de informações, com diversas observações diferentes, quanto mais diversificado os dados, mais chances de alcançar resultados satisfatórios. Além da diversidade, a quantidade de iterações e repetições (épocas) a serem feitas no treinamento de um algoritmo de rede neural tambem é de extrema importância. O número de iterações define quantas vezes os neurônios irão se relacionar entre si durante o treinamento do algoritmo para definir os pesos, e o número de épocas define quantas vezes será repetido as iterações, é a partir das épocas que podemos ver a evolução do algoritmo a cada treino realizado.

Neste trabalho, o conjunto de treino que contém 80% dos dados foi usado para ser treinar o algoritmo. Foram definidas 30 épocas e 28 iterações, apresentando uma boa curva de aprendizado durante as épocas, e nos indicando que aproximadamente 12 épocas já bastariam para alcançar valores satisfatórios obtidos pelo algoritmo, tanto em treino quanto em teste, nos apresentando uma acurácia de 94%.

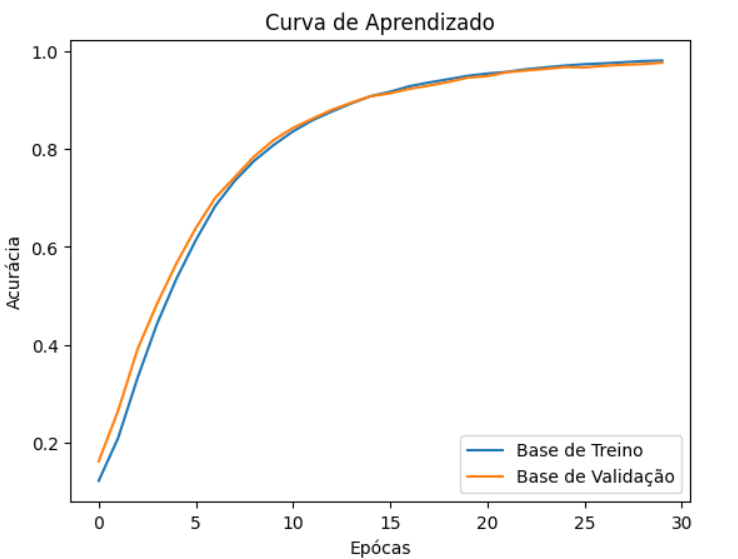


Figura 12. Curva de Aprendizado da Rede

Fonte: Resultados Originais de Pesquisa

Ao realizar a validação após o treino do algoritmo, usando a conjunto de teste, composto pelos 20% restante da base de dados inicial, foi obtido uma acurácia de 97% e perda de 7%. É imprescindível analisar os resultados obtidos após uma validação, pois nesse conjunto de resultados é possível obter informações que são de extrema relevância para julgar a maneira que o algoritmo está se comportando no momento de realizar as predições, podendo perceber as particularidades nos erros e acertos ao analisar cada algoritmo. Uma técnica que mostra esses resultados é a “Matriz de confusão”, é uma técnica de análise preditiva que cruza os resultados preditos com os resultados verdadeiros, retornando uma tabela que relata o número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos, auxiliando na verificação da existência de um “overfitting” no modelo desenvolvido, que são modelos que apresentam acurácia baixa quando são utilizados em dados fora da base de treino. Abaixo, na figura 13 podemos ver a matriz de confusão.

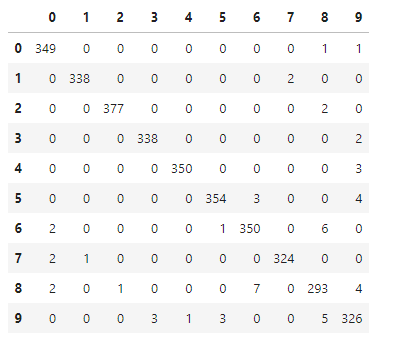


Figura 13. Matriz de Confusão

Fonte: Resultados originais da pesquisa

A matriz de confusão acima mostra a intersecção dos valores de referência (números de 0 a 9 na horizontal) com os valores preditos pelo modelo (números de 0 a 9 na vertical), utilizando a base de teste de dados, que foi explicada em “Materiais e Métodos”. A partir desses valores de cruzamento, vemos poucos erros na predição do modelo, erros esses que se concentram nos dígitos maiores do que 5, podendo afirmar que esses dígitos são confundidos entre si ao realizar a predição, sendo os dígitos 0, 6, 8 e 9, em determinadas observações acabou confundindo esses dígitos entre si e realizando uma predição incorreta, pode-se perceber isso na matriz de confusão. No entanto esses erros não comprometem a acurácia do modelo desenvolvido, pois mesmo com esses possíveis erros ao realizar a predição, o modelo apresenta ótimos valores de acurácia para cada digito, e por fim, é possivel afirmar que o modelo não sofre de “overfitting”, já que esses ótimos resultados foram obtidos da matriz de confusão que utiliza uma base de dados diferente da base de treino. Abaixo é mostrando a precisão do modelo para cada digito.

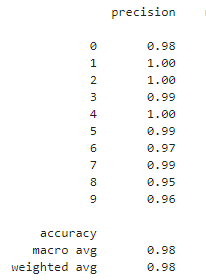


Figura 14. Precisão da Matriz de Confusão

Fonte: Resultados originais da pesquisa

Este trabalho além de apresentar a modelagem de um algoritmo de reconhecimento de dígitos de displays de 7-segmentos utilizando redes neurais convolucionais, uma técnica muito interessante e eficaz para esse tipo de tarefa e vários outros dentro da área de processamento de imagens, também apresentou a utilização na prática das técnicas da área de visão computacional. E ao combinar essas duas áreas, foi obtido resultados que podem otimizar inúmeros processos.

A visão computacional junto com o processamento de imagem, apresentam um horizonte de ferramentas e aplicações diferentes, coisas que antes eram um problema de se realizar visualmente, ou trabalhoso, pode ser realizado com poucas linhas de comando e de maneira muito mais eficiente.

**Considerações Finais**

O funcionamento de uma CNN torna ela uma ferramenta interessante, eficaz e poderosa, pois a partir da liberdade de construção de uma rede, nos é mostrado as inúmeras possibilidades de se modelar uma rede neural para diversos casos diferentes ou imagens diferentes, tornando sua aplicação ampla e satisfatória, é possível afirmar que a aplicação de redes neurais se tornou uma ferramenta desejável em diversas áreas, pois ela permite resolver problemas complexos e não-lineares, de uma maneira otimizada e confiável.

A base de dados auxiliou muito na construção de algoritmo que apresentou resultados excelentes, pois a sua variedade e extensão, propiciou uma realidade que é enfrentada ao implementar esse tipo de inovação, as aleatoriedades da vida real. Ao perceber que existem posições, intensidades de cor, espessura dos traços, inclinações e ângulos diferentes, é possível afirmar que esses fatores influenciam para a construção do algoritmo, e que levar em consideração a presença desses fatores apenas quando se é percebido durante sua aplicação, pode deixar o modelo “frágil” quando aplicado a esse tipo de aleatoriedade. Graças a essa particularidade desta base de dados, não foi necessário realizar a técnica de “data augmentation”, que consiste em realizar modificações que nos trazem esse tipo de aleatoriedade dentro da base de dados. Então, é possível afirmar que a aleatoriedade auxilia a construção de um modelo mais robusto.

A partir da implementação desse modelo em um sistema de captura de imagens combinado com as ferramentas de visão computacional, foi possível automatizar processos em que se era necessário a presença de uma pessoa para reescrever o digito lido, otimizando esse processo.

**Referências**

Alves, G.. 2018. Entendendo as Redes Neurais Convolucionais. Disponível em:<<https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>>. Acesso em: 20 dez. 2023.

Kumar, S.. 2023. Access Camera and Display Image Using Python and Opencv — the ELI5 way.

Disponível em: <<https://medium.com/@sunnykumar1516/access-camera-and-display-image-using-python-and-opencv-7e4b5d54375b>>. Acesso em: 24 dez. 2023.

Saha, S.. 2018. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. Disponível em: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>>. Acesso em: 20 dez. 2023.

Nicole, A.. 2023. Python e OpenCV: Explorando Threshold e Filtros. Disponível em: <<https://medium.com/@ana.nicole02/python-e-opencv-explorando-threshold-e-filtros-fc1289aa7115>>. Acesso em: 19 dez. 2023.

SuperAnnotate. 2023. Convolutional Neural Networks: 1998-2023 Overview. Disponível em: <<https://www.superannotate.com/blog/guide-to-convolutional-neural-networks>>. Acesso em 20 dez. 2023.

Monahan, C.. 2020. Like MNIST, but for 7-segment displays. Disponível em: < <https://connormonahan.net/2020/09/27/seven-segment-display-dataset.html>>. Acesso em: 26 out. 2023.

Rosebrock, A.. 2017. Recognizing digits with OpenCV and Python. Disponível em: < <https://pyimagesearch.com/2017/02/13/recognizing-digits-with-opencv-and-python/>>. Acesso em: 01 dez. 2023.

Paperwithcode. 2017. Disponível em: <<https://paperswithcode.com/method/max-pooling>>. Acesso em 22 dez. 2023.

Barros, M.. 2020. Prática 04 - BCD e Display 7 Segmentos. Disponível em: <<https://marcielbp.github.io/Circuits/lab/pr04.html>>. Acesso em 15 jan. 2024.