Hand Written Letter Classification using CNN

Aline Kurkdjian ID: 40131528 Lea Lakkis ID: 40125381

https://github.com/lealakkis/Comp478/blob/main/cnn-for-handwritten-letters-classification.ipynb

***Abstract*— Convolution Neural network is a deep learning model that has been used and is still used for image classification and recognition. Using deep convolutional neural networks can be more precise and accurate than using humans for image classification and recognition. This is due to the fact that machines can process larger sets of data than a human can. Classifying handwritten letters into their proper letter class is a multi-class classification problem as trying to read handwritten letters can be challenging for people in different fields. These fields include and are not limited to education, postal code services,forensics and document digitization. CNN would classify an image with a hand letter into the class of letter it belongs to. This will be performed by transfer learning using deep trained models from a large dataset on Kaggle.**

***Index Terms*— deep learning, transfer learning, neural network, convolutional neural network, image classification, convolution**

I. Introduction

Handwritten letter classification is utilised in many different categories for purposes such as document digitization, education, forensics and postal services. In document digitization, it could be used to digitise historical documents that are sometimes hard to read. In education it could help teachers digitise students handwritten assignments as some handwritings are harder to read than others. In forensics it could be used in criminal investigation that requires document analysis. For postal code it could be used to digitise handwritten addresses on letters and package labels. In all these areas handwritten letter classification is a way to enhance accuracy, efficiency and automation in various settings.

CNN systems are built to mimic the human visual system but can be more efficient as they can process large data sets in a smaller period of time than a human and therefore resulting in more accuracy and efficiency. Maximum validation accuracy found by using a simple neural network model was found to be 97%, and a CNN model is able to get 98% and more with a single convolution layer [1]. As for efficiency, CNN is efficient because images usually have high dimensionality but CNN is able to reduce the high dimensionality of an image without losing any information.

However, the system used in this project is not perfect as it can only classify one letter at a time but not a whole document. Also, It has limited data

because every human has a unique handwriting, hence a lot of variability. Furthermore, putting all the different types of handwriting in the dataset could take years. An alternative is using transfer learning using pre-trained models such as MNIST which has a larger dataset about handwritten letter classification. Data augmentation could also be used by artificially generating new data using the data that we already have to make the dataset larger.

Deep learning is a machine learning technique that trains computers to mimic natural human habits. A computer model gains an understanding to perform classification tasks straight from images. In deep learning models, the models can reach a higher level of accuracy than human-level performance. The models are trained by utilizing neural network architectures that have multiple layers and by a large set of labelled data. Deep learning matters since it has tremendously improved to the point that it outperforms humans in certain tasks such as classifying objects and letters in images. There are two main reasons that deep learning has lately been more useful. In deep learning large amounts of labelled data is needed. Also, a considerable amount of computer power is needed in deep learning. High-performance GPUs have a parallel architecture that is useful for deep learning. The combination of cloud computing or clusters lets development teams minimize training time for a deep learning network immensely.

The framework used for handwritten classification for this project is Keras. Keras is built on top of Tensorflow. Keras is an industry-strength framework that can scale to large clusters of GPUs. Tensorflow is developed by Google and it is the most popular deep learning framework. It is an end-to-end open source platform for machine learning.
It is important to have a large dataset for handwritten letters classification because it enables the CNN to have better accuracy from the various data that is being used. While having a large variation of different types of letters with different size, readability and style it is able to recognize more easily the different representations of the letters. The big dataset that is used for this project is from Kaggle.

A convolutional Neural Network (CNN) is the most widely used type of deep neural networks. CNNs have two parts which are a pooling mechanism that segments the image into features to be analysed. Also, a fully connected layer that takes the output of pooling and predicts the greatest label to describe the image.

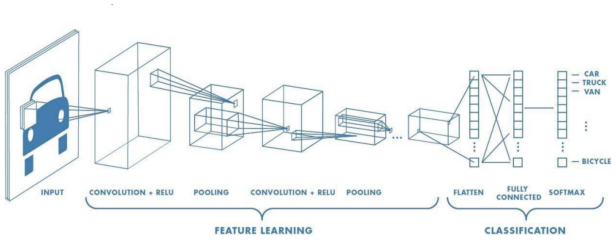

Fig. 1. [8]Network with many convolutional layers

This paper will go in depth about how many different image processing steps that are utilized to classify handwritten letters.

II. Related work

As hand-writtings have existed since very long time ago, from medieval time until now, sometimes especially when looking at historical documents,

only a hand full of people are able to translate those documents into more readable document. Furthermore, this type of technology is still not 100% reliable even if its used in many areas such as detection of traffic light signal, language translation and document extraction because the datasets that exist are not too big. Because of that, many people have taken interest in the study of CNN and the classification of hand written letters.

In a paper Convolutional-Neural-Network-Based Handwritten Character Recognition: An Approach with Massive Multisource Data, they talk about custom tailoring a twelve CNN model with two datasets for example Kaggle and MNIST which are proven to be lightweight and result is high accuracy. The custom tailored CNN models will include four convolutional layers, three max-pooling layers and two dense layers. Different otimizers and learning rates are then used to evaluate the best model out of the twelve.[2]

Furthermore, another paper Artificial Paleography: Computational Approaches to Identifying Script Types in Medieval Manuscripts where two machine learning methods VGG16 and FAU, a deep script model, were used to analyze medieval documents. In FAU data augmentation was used. The hand writing style identification had a 100% rate. [3]

III. Proposed Methodology

In order to classify handwritten letters to its proper class, the model will be trained using a dataset found on Kaggle composed of 3 .csv files. The 3 files are concatenated together and shuffled. Then processing is performed and one hot encoding is performed which is a technique used to convert categorical variables into a form that helps machine learning algorithms to provide a better prediction. These images are then passed into a CNN model for classification and then validation.

A. Dataset

The dataset that is utilised for this project is coming from Kaggle. There is one directory in this project which is Classification of Handwritten Letters. This directory is the combination of three folders which are called letters, letters and letters3. The three letters folders contain images of letters. To be precise, letters contain 1500 images, the letters 2 1650 folder contains and letters3 folder contains X images. In the dataset the setup is done in four parts which are dependencies, importing the data, concatenating the data and shuffle data.

In the dependencies section of the code it is importing all the needed libraries to work with the images, data and deep learning models. Some of the imported libraries are ‘pandas’, ‘numpy’, ‘os’, etc.The second step is importing the data. In this section is setting up the path to the directory where the image data is found. Also, it contains the csv files which contain the labels of the images.The third step is concatenating the data. In this step it is importing the label data from the csv files and concatenating them into a singular dataframe. In the dataframe it contains the information concerning the image files.

Fig. 2. [7]Handwritten latter

B. Processing

To make the most out of the datasets used, they will be processed through the processing layer. The first part of the processing payer is the one hot encoding. The data variable contains all the set of images from the dataset imported in the previous section. Is to use the head() function from the pandas library that returns the first few rows of a dataframe in our case the data variable that contains all the dataset. Then we concatenate

	letter	label	file	background	source
6491	щ	27	27_212.png	2	letters2/
2012	в	3	03_53.png	2	letters2/
13295	ы	29	29_336.png	3	letters3/
11270	с	19	19_311.png	3	letters3/
12837	щ	27	27_278.png	3	letters3/

Fig. 3. [7]Results with label and letter from first few rows of dataframe

all unique letter labels from the dataframe into a string called letters. We then process two columns in the dataframe called encoded_letter and encoded_background that will contain the values in the background and letter columns from the table above and convert it into a one-hot encoded numpy array.

We then proceed to create a variable called

label	file	background	source	encoded_letter	encoded_background
27	27_212.png	2	letters2/	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.0, 1.0, 0.0]
3	03_53.png	2	letters2/	[0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.0, 1.0, 0.0]
29	29_336.png	3	letters3/	[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.0, 0.0, 1.0]
19	19_311.png	3	letters3/	[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.0, 0.0, 1.0]
27	27_278.png	3	letters3/	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[0.0, 0.0, 0.0, 1.0]

Fig. 4. [7]Results table with encoded_letter and encoded_background

images where w store store all images in the data dataframe that are of size 32x32x1. If the images are not of that size then they will not be added to the images list. The list images is then normalized by dividing each pixel values by 255 which will scale the values of the pixel to [0,1]. Scaling the pixels usually helps the performance in deep learning. The final step of this process is to split the images list creates into a training set and a validation set.

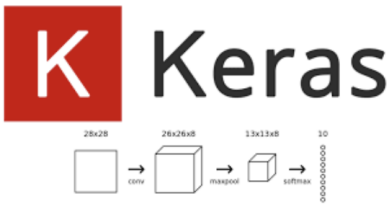

Fig. 5. [4]Keras Framework

C. Dependencies

In this project, the dependency that is being used is Keras. Keras is a framework that is built on top of Tensorflow. Tensorflow is developed by Google and it is the most popular deep learning framework.The major benefit of Keras is that it's a great framework since it can parallelize work with data. This means when keras is working with a large dataset such as this project, it greatly speeds up the learning models. Moreover, this framework uses python which is very

user friendly to use because it is easily understandable.

D. Model Architecture

CNN convolves learned features with input data, and uses 2D convolutional layers. CNN has two main part which are the convolution/pooling mechanism that breaks up the image into features and analyzes them while the other part is the fully connecter layer that takes the output of the convolution/pooling and predicts the best label to describe the image [8] In the CNN layer we will

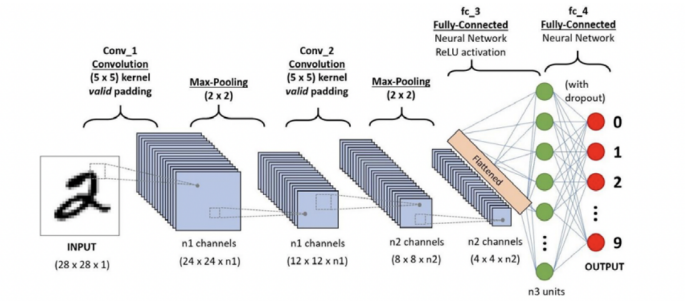

Fig. 6. [8]Convolutional Neural Network

user convolution, pooling, and flatten layer to create our model. We will also be using LeakyRueLu and callbacks[7].

A convolution layer is the process of passing a filter to an input which results in activation. The filter the image using a smaller filter in order to decrease the size of the image without decreasing the relationship between its pixels

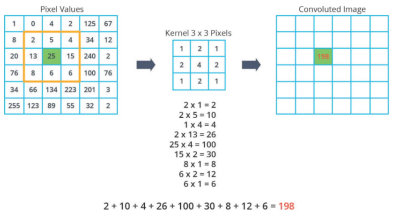

Fig. 7. [6]Convolution

A pooling layer is the process often happens after a convolution and is used to reduce the spatial size of the presentation in oder to reduce parameter count which in turn reduces computational complexity.

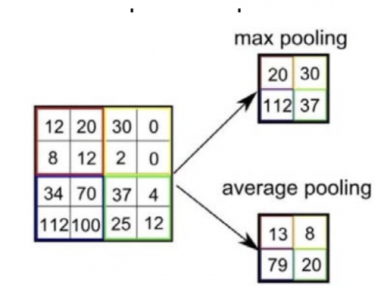

Fig. 8. [8]Pooling

The flattening layer only flattens the input from 2D to 1D tensor.
Dropout is used to reduce overfitting which randomly drops out some neurons at random during training.

LeakyRelu is an activation function that is used to not have vanishing gradient which could sometimes be a side effect of RELU. Vanishing gradient is when a large number of neurons die and stop learning because their output is zero[5]

Finally, callbacks is the process of calling a function after finishing excecutting another. This is useful to define a stopping criteria for when the model doesn't learn anymore through epochs

The first two steps with Convolution Neural Network (CNN) is defining the parameters and defining the custom metric. In the definition of the custom metric there is a function called top_3_categorical_accuracy() to evaluate the performance of a classification model.

The next step is to create the model. In this section it defines a CNN model using Keras for the cause of image classification. The CNN model takes four arguments which are the activation function, optimizer, evaluation metrics and loss function. The model architecture has many convolutional layers with various numbers of filters. This is continued by a max pooling layer to minimise spatial dimensionality. Also,dropout regularisation is implemented after the dense layer in order to prevent overfitting. Lastly, the compiled model is returned from the function.

The second part of this code is that it trains the CNN model with the use of the training data. On the validation data it validates the performance. Also, at each iteration the images are processed. In the training, using the specified metrics the performance on the validation set is monitored. Moreover, if the validation loss improves then the model weights are saved at each epoch. The ReduceLROnPlateau is utilised to minimise the learning rate of the optimizer. The ModelCheckpoint is utilised to preserve the weights of the best performing model with the help of validation loss. EarlyStopping is used to terminate the training process in the early stage if the validation loss has not been better for a reasonable amount of epochs. Lastly, the training process if placed in the object history, This contains the metric values and the loss for both validation and training sets above the epochs.

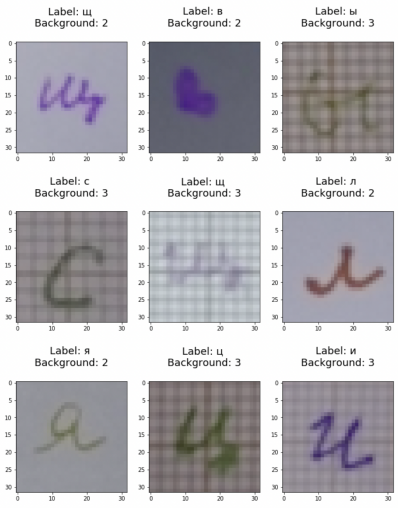

Fig. 9. [7]Results

E. Results

We obtained results by running the model with the list of images without taking into account the background recognition and then repeating the CNN model using background recognition.

The Results obtained by running the model without background recognition resulted in accuracy of 85.58% a loss of 0.5362 after 131 micro seconds. The graphs below shows the model accuracy graph as well ad the model loss graph The Results obtained by running the model without background

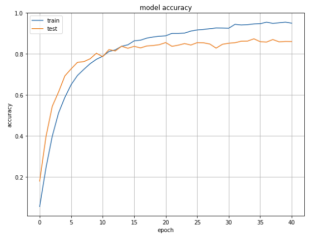

Fig. 10. [7]Accuracy graph for model without background

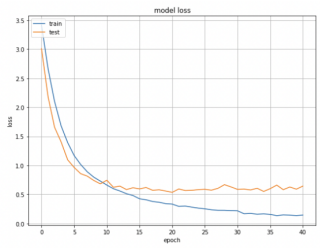

Fig. 11. [7]Accuracy graph for model without background

recognition resulted in accuracy of 99.65% a loss of 0.0119 after 148 micro seconds. The graphs below shows the model accuracy graph as well ad the model loss graph

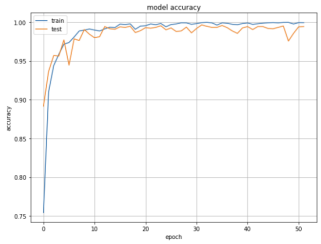

Fig. 12. [7]Accuracy graph for model with background

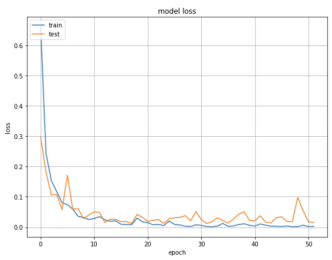

Fig. 13. [7]Accuracy graph for model with background

IV. Conclusion

To conclude, by the use of convolutional neural networks(CNN) in combination of the dataset retrieved by Kaggle and the use of the framework of Keras it demonstrates that these tools enhanced the accuracy. Based on this report, it is evident that this is a reliable method and proves to have great results with the use of background recognition.

The success rate of the accuracy is 99.65% of handwritten letter classification with CNN. This means that it has great performance and it is an efficient and reliable method for automating handwritten letters.

V. References

[1]M. Sanad, “Image classification using CNN (Convolutional Neural Networks),” Analytics Vidhya, 27-Apr-2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/#: :text=Even%20though%20our%20max%20validation,just%20a%20single%20convolution%20layer! [Accessed: 28-Apr-2023].

[2]Saqib, N., Haque, K. F., Yanambaka, V. P., & Abdelgawad, A. (2022, April 14). Convolutional-neural-network-based handwritten character recognition: An approach with massive multisource data. MDPI. Retrieved April 29, 2023, from https://www.mdpi.com/1999-4893/15/4/129

[3]Tarh, P. M. (n.d.). SSH, a secure protocol for managing and accessing remote servers. - — PMT. Retrieved April 29, 2023, from

http://www.payeshgaran.co/en/contents/news/1613538376966/-

[4]P. M. Tarh, “SSH, a secure protocol for managing and accessing remote servers,” - — PMT. [Online]. Available: http://www.payeshgaran.co/en/contents/news/16135 38376966/-. [Accessed: 29-Apr-2023].

[5]Papers with code - leaky relu explained. Explained — Papers With Code. (n.d.). Retrieved April 29, 2023, from https://paperswithcode.com/method/leaky-relu#: :text=Leaky%20Rectified%20Linear%20Unit%2C%20or,is%20not%20learnt%20during%20training.

[6]Balachandran, S. (2020, March 21). Machine learning - convolution with color images. DEV Community. Retrieved April 29, 2023, from https://dev.to/sandeepbalachandran/machine-learning-convolution-with-color-images-2p41

[7]Bryanb, “CNN for Handwritten Letters Classification,” Kaggle, 07-Nov-2020. [Online]. Available: https://www.kaggle.com/code/bryanb/cnn-for-handwritten-letters-classification/log. [Accessed: 29-Apr-2023].

[8] A. Ben Hamza, “Image Processing Slides.”