Implementation of CNN from scratch for Hand Gesture Recognition

Nishanjan Ravin[*], Parthan Olikkal[†] and Ambrose G.Tuscano[‡]

Group 3

Department of Computer Science, University of Maryland, Baltimore County

April 2020

Abstract

The goal of the project is to train a model, using the convolutional neural network machine learning algorithm, to be capable of recognizing different hand gestures, such as a closed fist, open palm, victory sign and others. The computer will have to ”learn” the features of each gesture and classify them correctly as far as possible. We aim to implement this system from scratch in python, without using any external deep learning libraries. The dataset creation and pre-processing will also be done manually. We will then observe our CNN's performance, with metrics such as accuracy and F1-score, and compare this result with the performance of other CNN models in similar applications, to evaluate our CNN.

Keywords— Convolutional neural networks, Hand gesture recognition, Pre-processing

[*] nishanr1@umbc.edu

[†] polikka1@umbc.edu

[‡] atuscan1@umbc.edu

1 Introduction

Hand gestures are a primitive form of human communication and one of the most natural form of expression. Evolutionary research suggests that the human language started with hand gestures and facial expressions, not sounds. Hand gesture recognition has a lot of potential applications, such as in sign language interpretation/translation or in interaction with machines. As hand gestures are increasingly considered to be superior in terms of convenience, many companies are trying to incorporate hand gestures as an option of providing inputs, rather than other complex forms of actions. However, the problem of hand gesture recognition has stifled progress in this field. In an attempt to solve this problem, many have turned their attention towards machine learning models.

While the neural network model is an effective form of machine learning, they suffer from two major disadvantages, particularly in the context of using images as the main dataset. Firstly, they take a lot of time to train, due to the rather large math computation required behind the neural network model. Secondly, they are particularly prone to overfitting, as they will only be able to consider the features of the images at the level of each individual pixel. Hence, in an effort to improve upon these two aspects of the neural network model, the **convolutional neural network** was developed.

Convolutional neural networks aim to use the spatial locality of images to extract useful features, which then aid in classifying the images more accurately. As such, several convolutional masks are constructed, such that the output of these convolutional masks with the images of the dataset produces feature-rich layers, which greatly assist and improves the classification ability of the model. As such, CNNs can also understand the complex and non-linear relationships amongst the images. Therefore, we plan on using a CNN-based approach to solve the hand gesture recognition problem.

The overall objective that we aim to complete in this project is to implement a working convolutional neural network, for the specific application of static hand gesture recognition. We aim to do this manually at all steps of the process, such as constructing our own dataset, conducting our own pre-processing, and developing the convolutional neural network by ourselves, without utilizing any 3rd party libraries or frameworks. By working on the whole process manually, we will be able to gain a better understanding of the working of convolutional neural networks and the process of utilizing an advanced machine learning model in the context of a real-world application, e.g. the requirements that need to met when creating a dataset and the corresponding pre-processing procedures. This will also give us a greater amount of flexibility, hence providing us with more room for experimenting with the convolution neural network model to find the right set of hyper-parameters that would optimize its performance.

2 Related Work

We went through a lot of related works on the topic of Hand Gesture Recognition, thus covering major implementations and notable research done in the field. Our aim while covering these works was to increase our understanding of the working of CNNs in the context of image classification (specifically hand gesture recognition), to enable us to prepare a suitable method of approach for our project.

Alex Krizhevsky's implementation on the ImageNet Data set for Classification with Deep Convolutional Neural Networks (4), was the one we traced back to, as it pioneered the application of the CNN structure to classify images in a supervised data set, and was considered to be groundbreaking research during its time. The authors of (3) provide a comprehensive survey on Hand Gesture Recognition systems, and also give insights on the structure that an ideal system needs to follow. The authors at (1) have written on the various techniques to implement Convolutional Neural Networks, and additionally, they have mentioned useful information such as various activation functions e.g. ReLU, max-pooling, etc. which we would be of great use in our manual implementation.

In (2), the ”data augmentation” procedure increased about 4% accuracy in the traditional CNN framework. The CNN model used here utilized a train to test split of 70:30, and achieved a maximum accuracy of 98.95%. The experimentation from this research would also help us create a better test and train split from the original dataset.

rES

Raimundo F. Pinto, et.al. at (9) works on how image pre-processing and subsequent segmentation to create a binary image or relevant and non relevant part (i.e. hand and background) affects the performance of the CNN. The research also compares the output produced when using various numbers of layered CNNs, and how they compare with the output obtained with the pre-built model constructed using the ”Keras” library. Here, a CNN of 4 layers showed the maximum precision of 96.86%.

Research at (6) worked on the process of data augmentation and utilizing dropout to reduce overfitting of the CNN model. By implementing ReLU as the primary activation function, and utilizing convolution and max-pool layers, they managed to attain a maximum accuracy of 88.5%. Our actual implementation of the CNN would take on a similar approach, but we will extend our focus to evaluating the performance by varying the number of layers and other hyper-parameter settings.

3 Methodology

We aim to manually create a data set of different gestures using different cameras and people for training the machine learning model. With a proper dataset in hand, our next step will be to work on pre-processing and normalization of the images of the dataset, so as to improve the input that we provide to the CNN, enabling it to perform better. The pre-processing step will be improved in parallel along with the development of the CNN, as there are several pre-processing steps (e.g. background subtraction, segmentation, etc.) which could be possibly incorporated in this step.

After that, we will create a simple neural network (fully connected single-layer) in Python and analyse its performance metrics on the dataset. Upon completing the simple neural network, we will work on adding a single additional hidden layer to the Neural Network (making it a fully connected multi-layer NN), and note down the improvement in its performance. After this step is completed, we can then add different types of hidden layers to the NN, such as convolutional layers and max-pool layers, to make a fully-fledged CNN.

The above step can be repeated multiple times, and after each modification to the CNN, we would be noting down the corresponding performance metrics. Finally, once we have obtained a proper CNN, we will experiment with different hyperparameter settings (e.g. activation functions, number of hidden nodes per layer, sizes of convolution masks, etc.) to observe the effect they have on the CNN, and to eventually reach optimal performance of the CNN.

In terms of the work that each group member needs to do, we have split up the workload by taking into account the relative difficulty of the different aspects of the project, while also considering which aspects of the project can be parallelized, so that efficient progress can be made. As the main two portions of the project revolve around pre-processing the data, and developing the actual CNN, we have divided the work as follows:

Parthan will be working on creating the dataset and then experimenting with several pre-processing techniques on the dataset, while Nishanjan and Ambrose will work on implementing the CNN, by first building a fully-connected single-layer NN, and then adding multiple various hidden layers such as convolutional layers and max-pool layers to make it a CNN.

4 Plan for Evaluation / Demonstration

Generally, the following metrics can be used in evaluating a CNN:

- Recall
- Precision
- Accuracy
- F1- Score
- Training time

As the F1-score gives us a weighted average of the recall and precision values, using it as a basis of measure would give us the best idea of the overall performance of the CNN. Thus, we plan to compare the F1-Score, accuracy and training time of the final CNN to its previous versions to observe the respective improvements, giving us an approximate measure of the level of improvement achieved at each revised version of the CNN.

Additionally, we will also compare the accuracy and F1-score measure of our CNN with other projects that are related to hand gesture recognition and see how we have fared in comparison.

Finally, we can build a separate CNN using any popular deep learning library, and create an equivalent CNN using these libraries and observe its performance on our dataset. This should give us a reasonably good idea of the performance of our manually developed CNN in comparison to the CNN which is implemented with libraries.

5 Objectives / Milestones

5.1 Main Objective

The overall objective of our project is to understand the concepts and the hurdles encountered when implementing a CNN from scratch, particularly in the context of a real world use, such as Hand Gesture Recognition. This would allow us to fully comprehend the various concepts and steps required to build the CNN, such as the necessary pre-processing steps, calculating loss, gradient descent, backpropogation, etc.

5.2 Milestones

- Achieve simple pre-processing of the dataset - 4/12/2020
- Achieve proper pre-processing and normalization of the dataset (e.g. background subtraction, image segmentation) - 4/28/2020
- Implement a single-layered fully connected neural network, upon which further improvements would be made - 4/12/2020
- Add a single hidden layer to the fully connected neural network - 4/22/2020
- Add a single convolutional and max-pool layer to the previously developed neural network - 4/28/2020
- Add multiple convolutional and max-pool layers to the CNN - 5/03/2020
- Experiment with different hyper-parameter settings to achieve maximum performance based on metrics such as F1-score and training time - 5/08/2020
- Look at CUDA implementation for training the model on GPU if time permits - 5/15/2020

References

[1] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B. et al. (2017). Recent Advances in Convolutional Neural Networks. Retrieved 10 April 2020, from https://arxiv.org/pdf/1512.07108.pdf.

[2] Islam, M., Hossain, M., Islam, R., Andersson, K. (2019). Static Hand Gesture Recognition using Convolutional Neural Network with Data Augmentation. Retrieved 10 April 2020, from https://ieeexplore.ieee.org/document/8858563.

[3] Khan, R., Ibraheem, N. (2012). Survey on Gesture Recognition for Hand Image Postures. Retrieved 10 April 2020, from https://pdfs.semanticscholar.org/085d/4a026eb425ff87094857e3a0ad4324419468.pdf.

[4] Krizhevsky, A., Sutskever, I., Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved 10 April 2020, from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[5] Lin, H., Hsu, M., Chen, W. (2020). Human Hand Gesture Recognition Using a Convolution Neural Network. Retrieved 10 April 2020, from https://ieeexplore.ieee.org/document/6899454.

[6] Mohanty, A., Rambhatla, S., Sahay, R. (2016). Deep Gesture: Static Hand Gesture Recognition Using CNN. Retrieved 10 April 2020, from https://link.springer.com/chapter/10.1007/978-981-10-2107-7_41.

[7] Molchanov, P., Gupta, S., Kim, K., Kautz, J. (2015). Hand Gesture Recognition with 3D Convolutional Neural Networks. Retrieved 10 April 2020, from https://ieeexplore.ieee.org/document/7301342.

[8] Murthy, G., Jadon, R. (2010). Hand Gesture Recognition Using Neural Networks. Retrieved 10 April 2020, from https://www.researchgate.net/publication/224120227_Hand_Gesture_Recognition_using_Neural_Networks.

[9] Pinto Jr., R., Borges, C., Almeida, A., Paula Jr., I. (2019). Static Hand Gesture Recognition Based on Convolutional Neural Networks. Retrieved 10 April 2020, from http://downloads.hindawi.com/journals/jece/2019/4167890.pdf.