Deep XGBoost, a new model for image classification

Jonathan Azpur
York University
jonaac@eecs.yorku.ca

Abstract

Convolutional neural networks (CNN) have become one the most popular neural network architectures when it comes to image classification. CNN models have been acknowledged as outstanding feature extractors, but the traditional classification layers can fail to understand the extracted features. This paper examines the possibility of combining CNN with eXtreme Gradient Boosting (XGBoost), a highly accurate and efficient classifying algorithm. The suggested model would integrate CNN as a feature extractor with XGBoost as a recognizer to classify images based on the feature output from the CNN. We test the model on the CIFAR-10 databases with three different CNN structures, and the results show that with simple CNN architectures the CNN-XGBoost model outperforms traditional CNN model, but when we start working with more complex CNN structures (VGG16 and ResNet50) the hybrid model is not outperforming the traditional CNN.

1. Introduction

Image classification has been one of the fundamental problems in the field of image processing and can be considered the basis of other computer vision problems (image localization, segmentation, object detection, etc.). Over the last few years the introduction of deep learning and deep neural networks [4] has been essential to the advancement of computer vision and has produced continuous breakthroughs in image classification [6][7].

One of the biggest challenges with image classification is the extraction of quality features from the original data sources. Convolutional neural networks (CNN) are a deep learning model capable of automatically performing feature extraction to obtain high quality features from the original sequence of data, this has made CNNs one of the most popular neural network architectures for image classification [15].

The architecture of a convolutional neural network can be described as a combination of two components. First, we have the feature extraction section, which is comprised of the convolution and pooling layers. Then, we have the classification section, which is normally comprised of a fully connected network and a softmax layer. On one hand CNN models have been acknowledged as outstanding feature extractors. But, on the other hand the traditional classification layers in the CNN can fail to understand the extracted features.

eXtreme Gradient Boosting [1] (XGBoost) is a machine learning algorithm built on the principles of the gradient tree boosting algorithm and designed for speed and performance. In recent years XBoost has become a very popular classifier due to it's efficienct and it's accuracy[2][11].

Given the limitation of the classification layers in a CNN model and the efficiency and accuracty of XGBoost, we want to examine if we can improve the performance of the traditional CNN structure by integrating a CNN with an XGBoost model. Our goal is to leverage the CNN to extract quality features and feed them as an input to the XGBoost to classify images.

The reminder of the paper is organized as follows. Section 2 will examine previous research related to this topic and describes the position of our project among the body of existing research. Section 3 provides some background, formal notations and definitions of the CNN and XGBoost models, followed by an overview of the proposed CNN-XGBoost model, the architecture of the different CNNs and the parameters selected for each hybrid model. Section 4 will provide a description of the experiments we developed to test our proposeed model, an overview of the CIFAR-10 databases, and a discussion of the experimental results.

2. Related Work

The CNN+XGBoost model has been researched and tested in the past few years. The model has been used as a predictior for protein subcellular localization [12], as a predictor for social media popularity [10] and there has been some basic testing of the model as an image classifier [13].

2.1. Protein subcellular localization prediction

Pang et al. [12] propose a new framework for protein subcellular localization prediction by using a CNN-

XGBoost model. According to the authors protein subcellular localization prediction is an essential task in bioinformatics and is essential in the further understanding the relationship among protein locations. After providing context on the current status of machine learning based methods used in this field, Page et al. identify that the current predictive models have issues extracting quality characteristics from the proteins and that the predictions performed fail to understand the relationship between sequences, and therefore see the opportunity for an integration between the CNN and XGBoost methods to tackle the shortcomings of the models used so far.

The proposed new CNN-XGBoost model uses a CNN with a similar structure to the LesNet-5 model proposed by Yann Lecun [9] consisting of two convolutional and pooling layers meant to obtain the proper feature representations, and they replace the fully connected neural network with XGBoost to predict the localization of subcellular of proteins. Page et al. test the model on four datasets containing protein sequences and they show that the proposed method achieves highly competitive performance.

2.2. Social media popularity prediction

Li et al. [10] propose the use of the hybrid CNN-XGBoost model to improve the predictions made on social media popularity (SMP). The authors identify the importance of time-scale features when it comes to social media analysis and the CNN as an adequate method for feature extracting from the more complex time-related data fields. The proposed hybrid model is composed of a CNN structure that is comprised of four convolutional layers, each followed by a pooling and activation layer and the convolutional layers feed into three fully connected layers. Given the high-level features extracted XGBoost performs a regression task on the popularity predictions. Li et al. test the proposed model on a real-world and publicly available dataset consisting of 432K Flickr images showing that the proposed hybrid model is capable of achieving competitive performance on the SMP task.

2.3. Image Classification

In 2018 Ren et al. [13] introduce the use of the hybrid model for image classification. Similar to the previous research on this field, the authors want to test if the hybrid model is able to outperform a traditional CNN. The proposed new CNN-XGBoost model uses the LesNet-5 architecture. Ren et al. test the model on the MNIST handwritten digital database and CIFAR-10 color image database, and they show that the deep XGBoost model outperforms the CNN model and the CNN-SVM hybrid model.

2.4. Novelty of this project

In terms of where my research falls among the body of existing body of work in this field, my focus on this project is to provide more detailed and complex testing to evaluate the actual validity of this model when it comes to image classification. Pang et al. and Li et al. focus on simpler data structures, in both cases the input to the model is a singular dimension matrix so we want to make sure that the CNN-XGBoost model can be extended to image classification. Even though Ren et al. have suggested the same model for image classification and their results have suggested it's competitive performance, I want to take this further and see how the proposed method compares to more complex CNN architectures.

In all the studies so far the CNN have been fairly simple in terms of structure, such that in two of the papers they use the LesNet-5 architecture which was introduced in the 90s and has been improved ever since. My goal is to build the hybrid structure with three different levels of complexity in terms of the CNN architecture and verify if the new proposed model is capable of a more accurate classification.

3. Methodology

In this section I will provide a brief description of convolutional neural netoworks and the XGBoost algorithm, and an introduction to the suggested CNN-XGBoot model. I will close out this section with a brief description of the dataset I will be using to test the accuracy of the proposed deep XGBoost model.

3.1. Convolutional Neural Network

The first work on modern convolutional neural networks (CNNs) occurred in the 1990s [9], inspired by the study of neocognitrons [3]. In the recent years it has become one of the most popular methods for image classification, big part of it's popularity in computer vision is due to its ability to automatically extract quality features from images through convolutional layers. A convolutional neural network is usually comprised of two main stages, first a feature extraction phase where we find combination of convolutional and pooling layers, then we have the classification phase where we have a fully connected layer. Figure 1 illustrates this structure.

The convolutional layer makes use of a set of learnable filters. A filter (kernel) is used to detect the presence of specific features or patterns present in the original image (input). This filter is convolved (slided) across the width and height of the input file, and a dot product is computed to give an activation map. Each convolutional layer can have multiple filters that detect different features and a set of activation maps is outputted.

The pooling layer is meant to reduce the number of

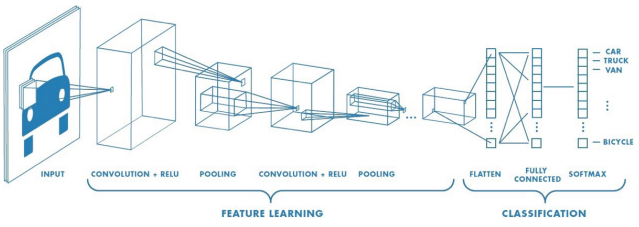

Figure 1. Overview of the CNN structure, composed by the feature learning layers and the classification layers

parameters and computational complexity of the network, therefore by decreasing the spacial size of the neural net it controls for overfitting. The pooling layer is usually placed in between convolutional layers. The combination of the convolutional and pooling layers allow the CNN to extract quality features from its input and feed them into the classifier.

The fully connected layer is composed of a fully connected neural network and receives the outputted values from the convolutional and pooling layers. This layer is meant to interpret the features extracted by the previous section and perform the classification task.

Convolutional neural networks have been the subject of interest for many researchers in the recent years and many state of the art architechtures have beeen proposed for image classification problems[14][5].

3.2. eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGBoost) was first introduced in 2015 [1] by Chen et al. XGBoost is a scalable end-to-end gradient tree boosting system, which is widely used to achieve state of the art results in machine learning tasks such as classification, regression and ranking. XGBoost introduce a novel system design (cache access patterns, data compression and sharding) to make the the algorithm scalable and efficient to deal with big data sets.

Boosting is a machine learning ensemble algorithm. Ensemble methods use multiple simple models and combine them to obtain a single strong model, in the case of boosting algorithms, it iteratively learns weak predictors with respect to a dataset and adds them to form final strong classifier. Gradient tree boosting (GTB) is a form of boosting that trains many weak CART (classificat and regression tree) models in a gradual, additive and sequential manner. Gradient tree boosting learns a set of weak trees by optimizing a cost function by iteratively choosing a weak hypothesis that points in the negative gradient direction. Let $\hat{y}_i^{(t)}$ be the prediction of the i-th instance, at each iteration t we want to minimize the cost of greedily adding a new tree learner f_t:

$$\mathcal{L}^{(t)} = \sum l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t) \qquad (1)$$

Where $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda||w||^2$ is an improved regularized objective introduced in XGBoost [1]. Let's define $q(x)$ as a CART tree and I_j as the instance set of leaf j in $q(x)$, by applying the Taylor second-order approximation to optimize the objective to general settings, removing the constraints and expanding Ω we can compute the optimal weight values w_j^* for a leaf j in $q(x)$:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \qquad (2)$$

and the corresponding optimal value of a fixed structure $q(x)$ can be calculated as

$$\hat{\mathcal{L}}^{(t)}(q) = \frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \qquad (3)$$

But, it is normally impossible to enumerate and calculate the cost of all the possible tree structures q, so instead the algorithm will greedly grow a tree from a single leaf, it will do so by calculating its optimal splits by using a cost function derived from 3.

$$\mathcal{L}_{split} = \frac{1}{2}[\underbrace{\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda}}_{\text{score of left child}} + \underbrace{\frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda}}_{\text{score of right child}} + \underbrace{\frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}}_{\text{score if split is not made}}] - \gamma \qquad (4)$$

In each iteration, using equation 4, the gradient boosting algorithm will greedily build a tree structure that minimizes the cost of miss-classification. XGBoost is a scalable and efficient implementation of the described gradient boosting algorithm, with improvements on it's regularization, and the addition of sparsity awareness for datasets with missing data and weighted quantile sketch to enable candidate split point generation among weighted datasets.

3.3. CNN-XGBoost

In this paper we suggest the integration of CNN models with the XGBoost algorithm to generate a single hybrid model for image classification. I will provide a brief description of the proposed model. First, the CNN will receive as input a normalized three dimensional matrix (Height x Weight x RBG) that represents the pixel values of an image, then once the CNN has been trained through backpropagation for image classification XGboost will replace the fully connected neural network in the CNN. So, XGBoost will be feed the features extracted by the trained convolutional and pooling layers of all the training images and it it will be trained to identify said features and classify the images accordingly. Once we have trained the CNN-XGBoost model

Model	eta	n_boost_round	max_depth
CNN-XGBoost	0.05	400	12
VGG16-XGBoost	0.05	100	24
ResNet-XGBoost	0.3	175	12

Table 1. Parameters for the XGBoost classifier on each CNN-XGBoost model

we can test it with new observed images. The goal is for our model to obtain quality features from the images and provide a more accurate classification than the traditional CNN architecture.

The CNN-XGBoost model has been proposed before by Ren et al. [13] and they show that the novel hybrid model is capable of of classifying images in the CIFAR-10 dataset more accurately than the CNN, CNN-SVM and XGBoost models. We want to expand on this field by testing the CNN-XGBoost models on more complex and accurate CNN architectures. First, I will build a CNN that can obtain results as good as the baseline (82% to 89% accuracy) proposed by the University of Toronto where one can download the CIFAR-10 database[1] which already performs 6 to 13% better than the LesNet-5 model used by Ren et al. I will also be using the VGG16 architecture [14] and the ResNet [5] architecture. I will be integrating all three CNN structures with an XGBoost and compare their performance to their original CNN architectures and the CNN-SVM/kNN hybrid models (with its corresponding CNN architecture).

3.3.1 Baseline

Shallow CNN architectures such as LesNet-5 were able to achieve an accuracy of 76%. So, the idea is to builds a deeper architecture that can better interpret features. With multiple layers a CNN can learn features at various levels of abstraction. From recognizing basic features like edges to shapes and finally objects. Multiple layers are much better at generalizing, but with deeper layers we also need to be careful to not over-fit the training data. Deep CNN structures has higher data requirements and some form of regularization.

The baseline CNN structure that I built will be based on the baseline results proposed by the website where the CIFAR-10 dataset is found [1]. The model consists of six convolutional layers, with three max pooling layer, each one after every two convolutional layers and they are followed by a flatten layer and a dense layer with softmax activation of 10 nodes (one per class). I will apply image augmentation to to the training dataset to fulfil the data requirements of our deeper architecture and I will add dropouts between layers and batch normalization to deal with regularization.

[1]The dataset and the baseline accuracy results can be found at: `https://www.cs.toronto.edu/~kriz/cifar.html`

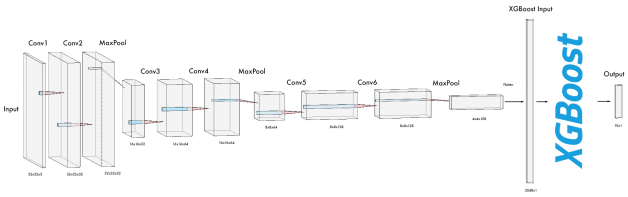

Figure 2. Structure of the VGG-16 model

When it comes to the CNN-XGBoost model we will replace the dense and softmax layers with the XGBoost algorithm. I will first perform hyperparameter tunning though cross validation on the XGBoost model with the features extracted by the trained CNN and finally, I will train the XGBoost model for image classification, Table 1 has the values of the parameters used for the CNN-XGBoost structure and Figure 2 is a visual representation of this architecture integrated with the XGBoost classifier.

3.3.2 VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman [14]. This model was able to achieve 92.7% accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

In terms of its structure, I had to make some minor changes to the input layer so it could be able to take in 32x32x3 sized images. The image is processed by a stack of convolutional layers, where the filters have a very small receptive field (3x3). The stride of the filters is fixed to 1 pixel, the padding of the input layer is such that the spatial resolution is preserved after convolution. Pooling is carried out by five max-pooling layers, which follow some of the convolutional layers. Max-pooling is performed over a 22 sized window, with a 2 pixel stride. The structure is finalized with three fully-connected layers (including the output layer). Figure 3 has a visual representation of the VGG16 architecture. In the case of the CNN-XGBoost model we will replace the the three fully connected layers with an XGBoost. I will first perform hyperparameter tunning though cross validation on the XGBoost model with the features extracted by the trained VGG16 and finally, I will train the XGBoost model for image classification, Table 1 has the values of the parameters used for the VGG16-XGBoost structure.

3.3.3 ResNet

Deep Residual Networks (ResNet) has been one of the most groundbreaking neural network architectures in the field of computer vision. It was introduced by He et al. [5] in 2015.

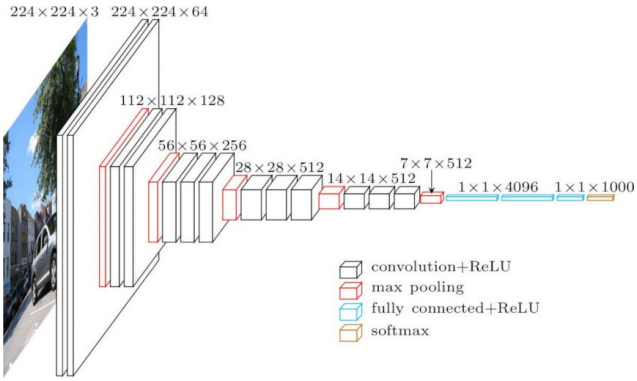

Figure 3. Structure of the VGG-16 model

In recent years CNN structures have been growing deeper and deeper, but increasing depth is not just as simple as stacking layers over and over. As networks go deeper, its performance gets saturated. ResNet is based on the idea of introducing identity shortcut connections between layers (skipping at least one layer at a time). He et al. argue that stacking layers shouldn't degrade the network's performance, thanks to the stack of identity (or residual) mappings on a network, which allows gradients to reach any earlier layer. So, deeper models should not perform worst than their shallower versions. The hypothesis is that allowing the layers fit a residual mapping makes it easier than directly fitting the desired mapping. Future studies showed 1001-layer deep ResNet could outperform shallower structures.

I will be using a ResNet structure of 50-layers including a fully connected neural network to perform the classification task. In terms of the CNN-XGBoost model I will be replacing the fully connected layers with the XGBoost model. I will first perform hyperparameter tunning though cross validation on the XGBoost model with the features extracted by the trained ResNet50 and finally, I will train the XGBoost model for image classification, Table 1 has the values of the parameters used for the ResNet-XGBoost structure.

3.4. CIFAR-10

The CIFAR-10 dataset is a labeled subsets of the 80 million tiny images dataset [2]. The dataset consists of 60,000 32x32x3 images of color that can be clasified in 10 categories, with 6000 images per class. There are 50000 training images and 10000 test images.

The training set contains exactly 5000 randomly-selected images from each class. The test set contains exactly 1000 randomly-selected images from each class. Figure 4 shows

[2]The dataset is available at: `https://groups.csail.mit.edu/vision/TinyImages/`

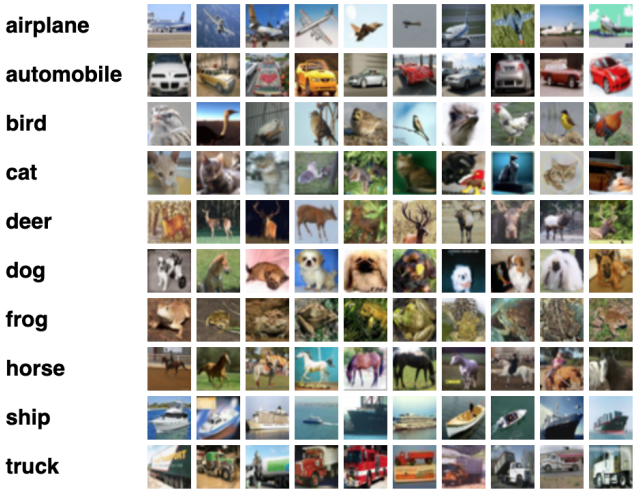

Figure 4. Structure of the VGG-16 model

all the classes in the dataset, as well as 10 sample images from each category.

4. Results

In this section we want to verify the potential validity and improvement of the proposed CNN-XGBoost model. We will be comparing the accuracy, for each one of the CNN-XGBoost architectures mentioned in Section 3, with its original CNN structure, and the CNN-SVM and CNN-kNN models. All tests will be run on the CIFAR-10 database.

4.1. Baseline

Figure 5 shows the exact accuracy of all the models. We can see that the CNN-XGBoost model has a higher classification accuracy (2% higher) than it's original CNN model and the difference id even higher with the other integrated models. These results seem to imply that the XGBoost model is better at interpreting the features extracted by the CNN and therefore can better classify images than the original fully connected layer.

4.2. VGG16

Figure 6 shows the exact accuracy of all the models based on the VGG16 architecture. Contrary to what we expected, the CNN-XGBoost model has a lower classification accuracy (only 0.3% lower) than the original CNN model. But, compared to the SVM and kNN integrated models the proposed model had a higher accuracy. It seems to show that with a more complex structure such as VGG16 the XGBoost is not capable to better interpret the features extracted by the VGG16's convolutional layers and therefore can't outperform the trained fully connected neural network. The reason probably lies in the fact that the VGG16 model has a more complex architecture for its fully connected layers

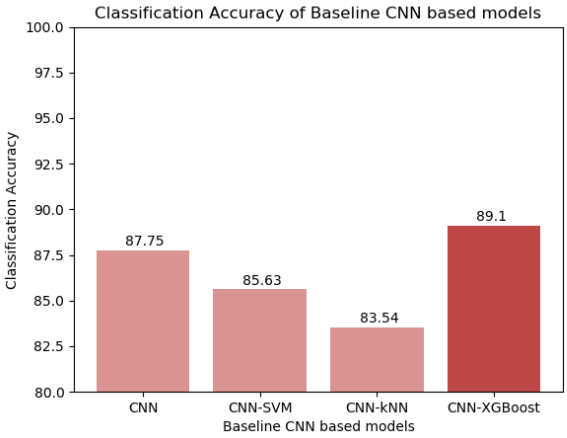

Figure 5. Classification Accuracy of Baseline CNN based models

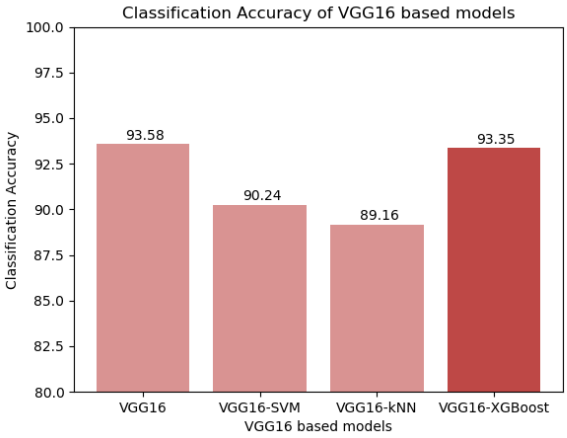

Figure 6. Classification Accuracy of VGG-16 based models

than the previous CNN structures used in these comparisons.

4.3. ResNet

Figure 7 shows the exact accuracy of all the models based on the ResNet50 architecture. Again, contrary to what we expected, the CNN-XGBoost model has a lower classification accuracy (4% lower) than the original CNN model. But, compared to the SVM and kNN integrated models the proposed model had a higher accuracy. Adding to the evidence found from the CNN-XGBoost model based on the VGG16 structure,it seems to show that with a more complex like ResNet the XGBoost is not capable to better interpret the features extracted by the ResNet's convolutional layers and therefore can't outperform the trained fully connected neural network. Simlar to VGG16, the reason probably lies in the fact that the ResNet model has a more

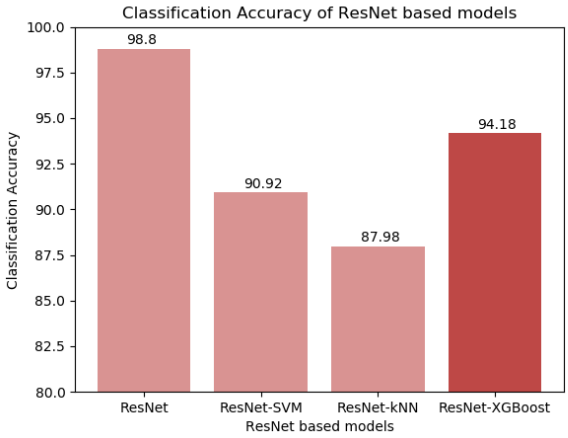

Figure 7. Classification Accuracy of ReNet50 based models

complex architecture for its fully connected layers than the previous CNN structures used in these comparisons.

5. Conclusion

The goal of this project was to examine the possibility of combining CNN with eXtreme Gradient Boosting (XGBoost), and leverage the strengths of each learner to produce a new integrated model that could outperform the CNN model in image classification. In order to thoroughly test this hypothesis I trained three CNN-XGBoost models, each one with a different complexities of CNN architecture (baseline, VGG16 and ResNet50), tested their accuracy on the CIFAR-10 database and compared their performance with its original CNN structure, and the CNN-SVM and CNN-kNN models.

In all three cases the CNN-XGBoost model was able to outperform the CNN-SVM and CNN-kNN models, and the higher the complexity of the CNN structure the bigger was the difference between our proposed model and the other two hybrid models. In the case of the baseline CNN model we saw that the CNN-XGBoost model was also able to outperform the original trained CNN model. But, once we started working with the more complex models both VGG16-XGBoost and ResNet-XGBoost had a lower classification accuracy than their original CNN models, the more complex the model the bigger was the difference in performance. More testing should be performed in this field but the results obtain lead us to believe that in the case of more complex CNN architectures (which also include more complex fully connected networks), the fully connected layers are better at interpreting the extracted features and the integration of the XGBoost model with the CNN do not necessarily lead to an improvement in image classification performance.

In terms of future work in this field, it would be interesting to see further testing with other CNN architectures and further testing with more complex image data sets. Also, we could introduce the CNN-XGBoost model to other fields of computer vision, such as image segmentation or object recognition, and given the recent popularity of CNN in speech recognition, it could be another interesting field to introduce the proposed model.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[2] Iyad Lahsen Cherif and Abdesselem Kortebi. On using extreme gradient boosting (xgboost) machine learning algorithm for home network traffic classification. In *2019 Wireless Days (WD)*, pages 1–6. IEEE, 2019.

[3] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[7] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *UK Workshop on Computational Intelligence*, pages 191–202. Springer, 2018.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[9] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[10] Liuwu Li, Runwei Situ, Junyan Gao, Zhenguo Yang, and Wenyin Liu. A hybrid model combining convolutional neural network with xgboost for predicting social media popularity. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1912–1917. ACM, 2017.

[11] Jun Ma, Yuexiong Ding, Jack CP Cheng, Yi Tan, Vincent JL Gan, and Jingcheng Zhang. Analyzing the leading causes of traffic fatalities using xgboost and grid-based analysis: A city management perspective. *IEEE Access*, 7:148059–148072, 2019.

[12] Long Pang, Junjie Wang, Lingling Zhao, Chunyu Wang, and Hui Zhan. A novel protein subcellular localization method with cnn-xgboost model for alzheimer's disease. *Frontiers in genetics*, 9, 2018.

[13] Xudie Ren, Haonan Guo, Shenghong Li, Shilin Wang, and Jianhua Li. A novel image classification method with cnn-xgboost model. In *International Workshop on Digital Watermarking*, pages 378–390. Springer, 2017.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Farhana Sultana, Abu Sufian, and Paramartha Dutta. Advancements in image classification using convolutional neural network. In *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pages 122–129. IEEE, 2018.

Appendix

User Manual

For each CNN structure (Baseline, VGG16 and ResNet50) I have developed a program to build the original CNNs (cnn, cnn_vgg and cnn_resnet), another program were I train the CNN-XGBoost model by loading the built CNN, drop the fully connected neural network, feed the features extracted tothe XGBoost and train the XGBoost (cnn_xgboost, cnn_vgg_xgboost, cnn_resnet_xgboost), and finally a program where I load the CNN, CNN-SVM, CNN-kNN and CNN-XGBoost models for each and compute their accuracies (accuracy_baseline, accuracy_vgg, accuracy_resnet).

System Design

All programs are developed in python. I developed in python 3.6 and I used the xgboost, keras, tensorflow, scikit-learn, sklearn, numpy, scipy and pickle.

Sample input and output

I will be using the CIFAR-10 dataset which is included with the report and the code under the /data directory