Introduction to Artificial Intelligence
Data Mining

Wiktor Łazarski
Computer Science student
Faculty of Electronics and Information Technology
Warsaw University of Technology

ABSTRACT

Image classification is a common task for a Computer Vision (CV) systems. It is very important aspect of Artificial Intelligence (AI) because it allows to implement some level of sense of perception in computers. Vision is a particularly important sense for every 'intelligent' creature. It allows to observe and analyse surrounding environment and based on those observation plan further actions. Therefore, a lot of efforts are put to somehow accurately model visual systems in computer. The goal of this project is to perform some researches how different learning algorithms perform in terms of a binary classification of RGB images and what kind of information can be mined using them.

Source code: https://github.com/wiktorlazarski/Doggos-vs-Cattos

1. INTRODUCTION

Classification and recognition has been researched for many years. Only few years ago, with increased popularity of Deep Neural Network (DNN), those problems got a huge boost when it comes to accurately recognizing objects in the images. Before, people tent to use other Machine Learning (ML) algorithms to classify objects. Those approaches results in poor accuracy, which does not allow to implement reliable vision software in many cases. In this report, as a project for Introduction to Artificial Intelligence course, I decided to compare standard ML algorithms – *k-Nearest Neighbors (kNN), Support Vector Machine (SVN), Random Forrest* – that people used to implement for classification problem with nowadays state-of-the-art Convolutional Neural Networks (CNN) with respect to accuracy defined by a formula:

$$Accuracy = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positive} + \text{False Negative}}$$

What I found out is that even without specially tuned hyperparameters and regularization of CNN, we can observe a huge jump in accuracy when it comes to classification task with respect to previously ML algorithm.

2. DATASET

The dataset used in my studies contains 25k images of cats and dogs divided equally. Therefore, it is even more than enough quantity of data for standard ML algorithms and sufficient quantity for CNN. All images are in various size so before executing learning procedure on every model, we must somehow preprocess those images by rescaling them to the same dimension. While doing preprocessing procedure for different algorithms we can already observe ML algorithms drawback regarding size of an image. It turned out that programmer must test different image sizes to properly adjust time of execution of predicting procedure.

Link to dataset: https://www.kaggle.com/c/dogs-vs-cats/overview

3. STANDARD MACHINE LEARNING ALGORITHMS RESULTS

3.1 k- Nearest Neighbors

To tune and find the best suitable hyperparameters for kNN model I used Grid Search (GS) method. Hyperparameters that result in the highest accuracy metrics for our classification problem are presented in a Table 1.

Hyperparameter	Value
k	5
comparison metrics	Minkowsky distance with p=2

Table 1: Best kNN hyperparameters

This model resulted with accuracy score, on test set, equals 54% and Receiver Operating Characteristic Curve (ROC) presented in Figure 1.

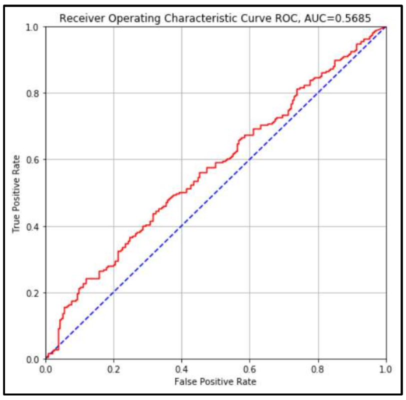

Figure 1: ROC of kNN model

As we can see model is completely useless. Accuracy being equal to 54% indicates that it is only 4% better than a model that would perform random guesses. Also moving probability threshold does not help what is shown in Figure 1.

3.2 Support Vector Machines

3.2.1 Linear Support Vector Machines

Linear SVM are a special kind of SVM that uses linear kernel when making predictions. This allows as to return model's coefficients and plot them to see filters of patterns that linear SVM is looking for. I train linear SVM on few samples (ten) and plot coefficients of few such trained models. The results are present in Figure 2.

Figure 2: Linear SVM trained on 10 samples filters

However, when doing the training on more samples filter is more noisy and such clearly visible patterns are no longer available. Linear SVM filter train on one thousand samples is presented in Figure 3.

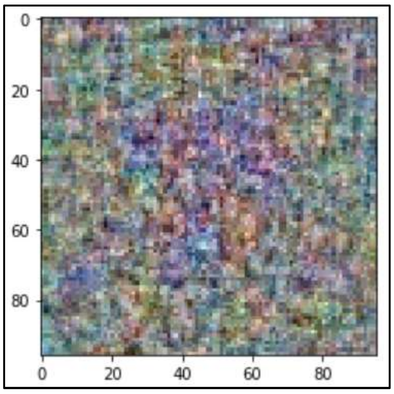

Figure 3: Linear SVM trained on 1k samples filter

We can see that more samples in a training dataset results in a more noise filter image which results in smaller accuracy. However, single SVM trained on small number of samples gives filter on which information that this filter is looking for is visible. Hence, maybe what can be done is to train a lot of SVM using few samples and involve some kind of Bagging method to make our predictions.

Linear SVM resulted in accuracy metrics equals 55%, which leads to the same conclusions as the one presented in kNN.

3.2.2 Non-linear Support Vector Machines

SVM with non-linear kernels where tested in my studies. To find the best suitable kernel I used the GS. Hyperparameters that result in the highest accuracy metrics for our classification problem are presented in a Table 2.

Hyperparameter	**Value**
C	1.0
kernel	Polynomial

Table 2: Best Non-linear SVM hyperparameters

Non-linear SVM resulted in accuracy metrics, on test set, equals 56%, which also does not satisfy any reasonable application.

3.3 Random Forrest

Random Forrest algorithm was and is very widely used in many ML problems. Even with huge impact of DL, we can still observe problems when Random Forrest models outperform DNN. To check how Random Forrest model will perform in terms of image binary classification I tune model with hyperparameters present in Table 3.

Hyperparameter	**Value**
Number of estimators	200
Max depth of a single Decision Tree	20

Table 3: Random Forrest hyperparameters

This model resulted with accuracy metric, on test set, equals 65% and Receiver Operating Characteristic Curve (ROC) presented in Figure 2.

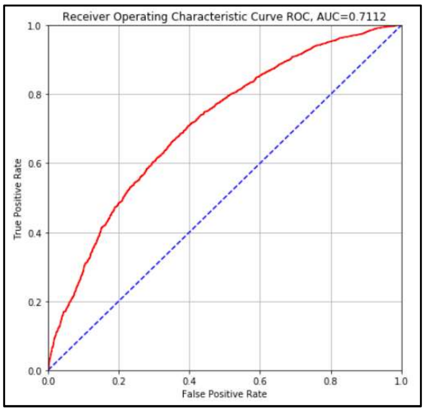

Figure 2: ROC of Random Forrest model

We can observe ~10% increase of accuracy with respect to previously examined models. However, still 64% is not sufficient accuracy to use this model in any application, because it is only 14% better than random guess classifier.

4. CONVOLUTIONAL NEURAL NETWORKS

CNN are called to be state-of-the-art when it comes to an image classification problem. It is a special type of DNN which is widely used in any image processing for AI because it prevents a special distribution of pixels. Previous models and standard DNN does not keep any relation between pixels position in the image. There is no doubt that this information is necessary for deciding what is present in the image. At first I proposed my architecture of CNN (present in Figure 3) and then I decided to research more standard approach, which is taken in CV task, meaning Transfer Learning (TL). TL is an idea that we take already trained DNN and we use it to extract feature available in a sample. In my study, I used VGG16 architecture to which I add densely connected layers to perform final classification decision. Architecture of this CNN is presented in Figure 4.

4.1 My Convolutional Neural Network Architecture

My CNN architecture resulted in accuracy score, on test set, equal to 83%. It is worth to notice that I did not put a lot of effort on finding and tuning proper hyperparameters like learning rate or optimization algorithm. I just followed my intuition when constructing the architecture and training the model. Even though, I resulted in a accuracy significantly bigger than previously. The accuracy gap between my CNN and Random Forrest model is 18% which can be seen as really significant difference.

4.2 Transfer Learning with VGG16 architecture

As mentioned above I also researched TL approach. CNN part of VGG16 architecture trained on ImageNet dataset was taken for my studies. Also as previously, I did not put a lot of effort to set proper hyperparameters. Even though, the obtained accuracy is equal to 90% on a test set. It is 7% increase of accuracy with respect to my CNN architecture.

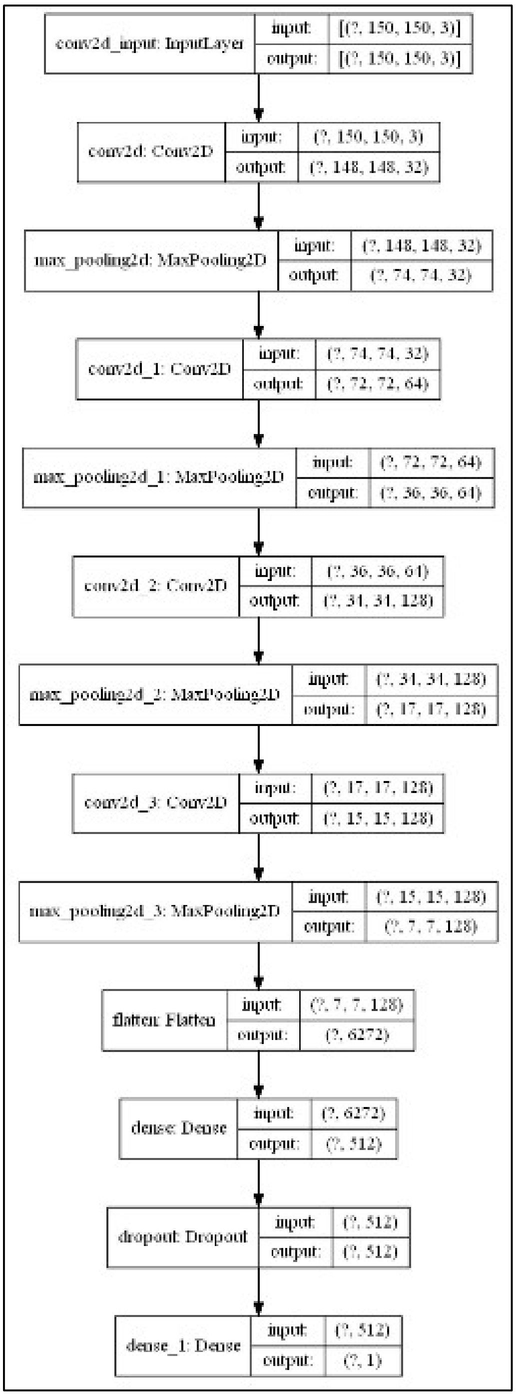

Figure 3: My CNN architecture

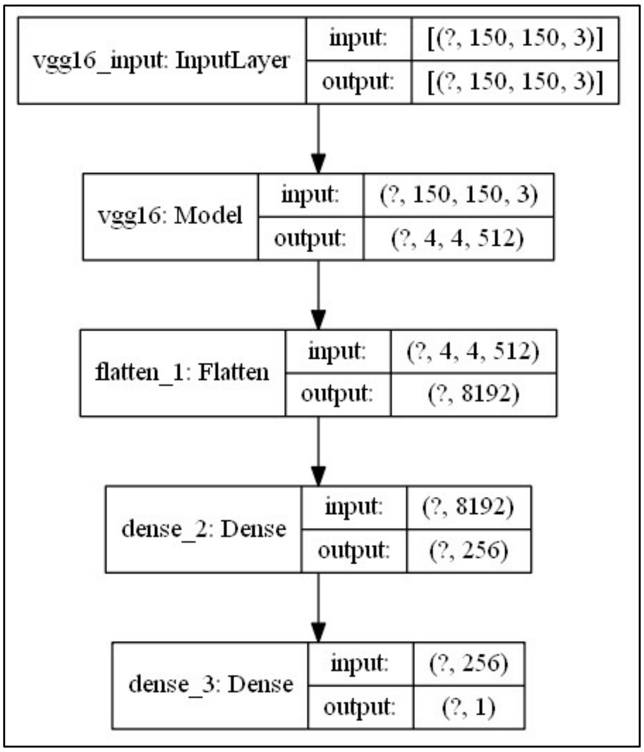

Figure 4: TL CNN architecture

5. CONCLUSIONS

In the report, I shown that currently called state-of-the-art model, CNN, for image classification is outperforming standard ML algorithms. In Table 4, the differences between obtained accuracy for different model are presented. It is clearly seen that CNN models outperforms standard ML models and should be used for every image processing for AI problems. Also it should be noticed that when describing CNN architecture I highlighted the fact about my low effort put to receive the best possible model. In my opinion, with time spend on finding proper hyperparameters and setting proper regularization technique like dropout, we may observe the ~5% of increase in accuracy. CNN are models that result in accuracy sufficient enough to produce CV systems that somehow rely on classification decision.

Model	Accuracy
kNN	54%
Linear SVM	55%
Non-linear SVM	56%
Random Forrest	65%
My CNN architecture	83%
TL CNN architecture	90%

Table 4: Comparison of models' accuracy

6. BIBLIOGRAPHY

1. "*Computer Vision with Deep Learning - lecture notes*" Stanford University

2. "*Deep Learning with Python*" François Chollet

3. "*Deep Learning*" Ian Goodfellow, Yoshua Bengio, Aaron Courville

4. "*Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*" Aurélien Géron

5. "*Python for Data Analysis*" Wes McKinney