CS 6375

Project Report

Hand Written Digit Recognition

Number of free late days used: 0

Paril Doshi - PSD170000

Vivek Shah - VXS173830

Debanjana Dasgupta – DXD170016

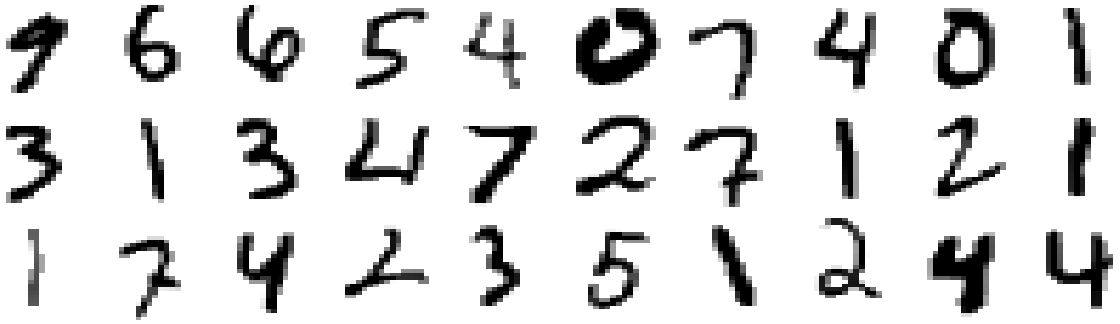
Contents

INTRODUCTION AND PROBLEM DESCRIPTION

The project is on Digit Recognition from the Image. The Image would a hand-written digit inscribed in it. The Dataset used is downloaded from Kaggle competition "Digit Recognizer".

https://www.kaggle.com/c/digit-recognizer

The goal of the project is to identify hand-written images of digits. The project is a classic example of classification algorithm in machine learning where there are 10 classes of digits (0-9). The solution involves creating a machine learning model which can predict the correct digit from the test image.



The dataset is in two parts 1) train.csv – 42000 Labelled Images

2) test.csv – 28000 Un-labelled Images

After training the model on train data the final output on test dataset is to be uploaded on the Kaggle website.

RELATED WORK

We took this project from the world-famous online data scientists and machine learning community, Kaggle. The purpose of choosing this project was to explore the combination of machine learning and computer vision. We wanted to work with images and this project is the "hello world" dataset of computer vision. As we were working on this project, we looked for references of related work and we found the following:

1. LeCun, Yann, et al. "Comparison of learning algorithms for handwritten digit recognition." International conference on artificial neural networks. Vol. 60. 1995.
   This paper compares the relative merit of several classification algorithms developed by Bell Laboratories and elsewhere for recognizing hand-written digits. They used the following classifiers in their study: Baseline Linear Classifier, Baseline Nearest Neighbor Classifier, Pairwise Linear Classifier, PCA and Polynomial Classifier, Radial Basis Function Network, Large Fully Connected Multi-Layer Neural Network, Tangent Distant Classifier and Optimal Margin Classifier.

2. Jonathan J. Hull, “A Database for Handwritten Text Recognition Research.” IEEE TRANSACTIONS ON PA’ITERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 16, NO. 5, MAY 1994.
   This paper describes an image database of handwritten text recognition research. It consists of 5000 city names, 5000 state names, 10000 ZIP Codes, and 50000 alphanumeric character. These images were scanned at 300 pixels/in in 8-bit grayscale which gives us an idea of the of how the image data is stored and how the preprocessing and grayscale identification is done experimentally.

After reading the related works we had some idea of the algorithms that will work well with this kind of dataset and which algorithms would give good result. For example, we had this intuition that neural network will be a good solution for this problem

DATASET DESCRIPTION

The dataset is of MNIST type.

The dataset contains 60000 images of hand-written digits. Out of which 42000 images are used for Training and other 28000 for Test.

There are following number of training samples available for each class:

Digit	Number of Samples
0	4132
1	4684
2	4177
3	4351
4	4072
5	3795
6	4137
7	4401
8	4063
9	4188

The training dataset has 785 columns. The first column is for labels and the other 784 columns are pixel values for a 28 x 28 image

The test dataset has 784 columns of pixel value for a 28 x 28 image.

Each pixel column in the training set has a name like pixelx, where x is an integer between 0 and 783, inclusive. To locate this pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27, inclusive. Then pixelx is located on row i and column j of a 28 x 28 matrix, (indexing by zero).

For example, pixel31 indicates the pixel that is in the fourth column from the left, and the second row from the top, as in the ascii-diagram below.

```
000 001 002 003 ... 026 027
028 029 030 031 ... 054 055
056 057 058 059 ... 082 083
 |   |   |   |  ...  |   |
728 729 730 731 ... 754 755
756 757 758 759 ... 782 783
```

The pixel value ranges from 0(Black) to 255(White) and middle values are for shades of grey color.

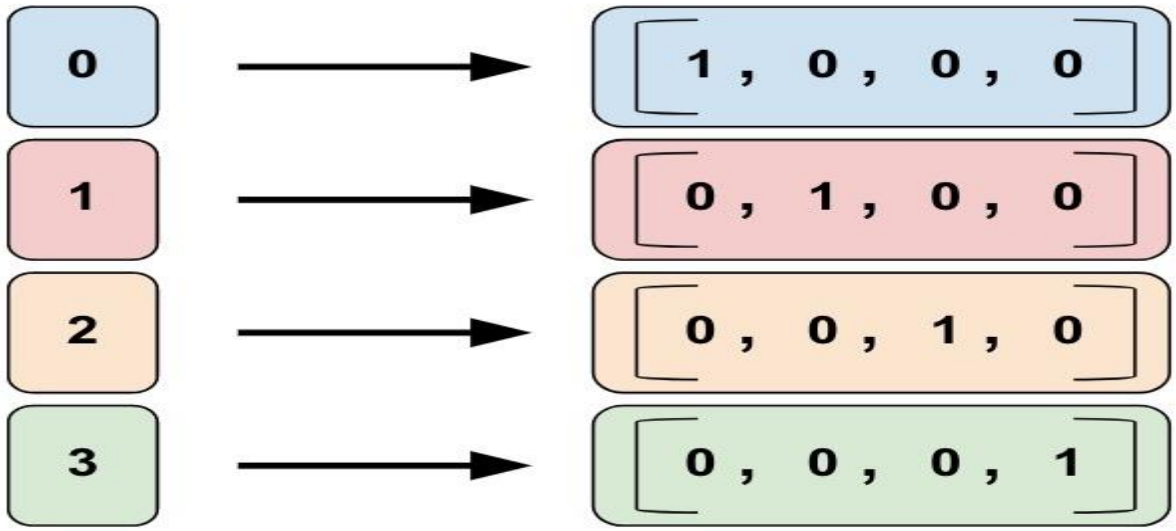
PRE-PROCESSING TECHNIQUES

The training data is a gray scale image with value range of (0,255). And the label are in the form of numbers ranging from 0 to 9.

For preprocessing we have divided all the pixel value by 255 and the image data is a 2-dimensional data and CNN needs a 3-dimensional data column as width, row as height, and depth as color channels. As the images are gray scale so the color channel is 1.

Image(28, 28) -> reshaped Image(28,28,1).

For output we have used keras library to convert single digit number to one hot encoding format.



PROPOSED SOLUTION AND METHODS

Convolutional Neural Network

In our project we have used Convolutional Neural Network for classification and found it the best among all.

The training data consisted of 48000 labelled images and testing data consisted of 12000 unlabeled images given in the Kaggle competition.

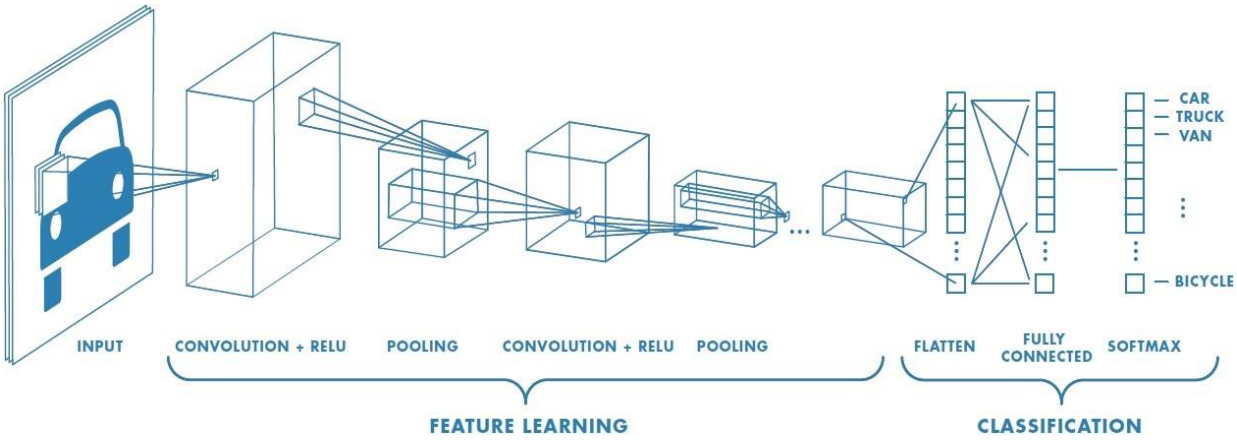
We had evaluated the result of CNN by two methods:

1) In first method complete training dataset was to train the model and then the unlabeled testing data was predicted, and the output was submitted online, and we got the accuracy of 0.99057. The link to the submission result is here - Link.

2) In this we had further divided the labelled data into training data and testing data for displaying the accuracy, loss, area under ROC, ROC curve and confusion matrix.

The details of each is shown in next section.

Convolutional neural network is a deep learning model capturing all the details of dataset in the form of weights in the network.



A CNN has 3 parts:

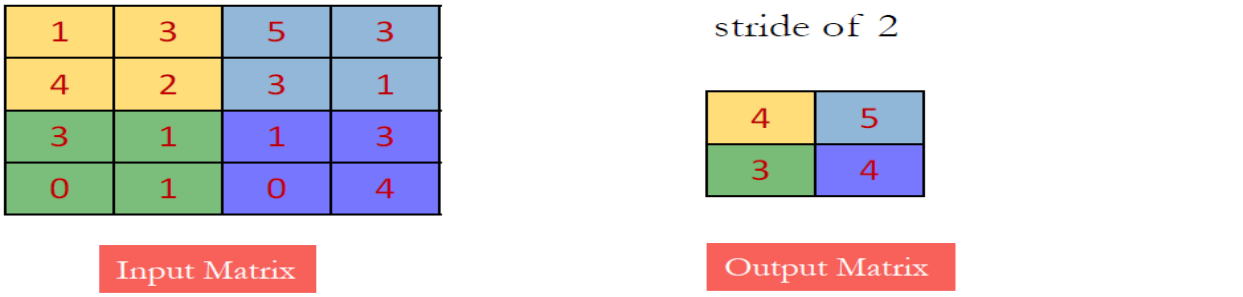
1) Convolution Network - Part of Feature Learning

This layer learns the relations between pixel values of an image and preserves it in form of weights in form of square matrix also know as filter matrix. In our project we have convolution network, all using ReLU as activation function and two having filters of size(3,3) and one having (2,2). This filter matrix is updated each time and it values basically represents the features of images and so also known as feature map.

For activation function we have used ReLU, as it captures all the non-negative values of real world in their original form(max(0,x)) while other functions scale the non-negative values between certain range and this decreases accuracy.

2) Pooling Layer - Part of Feature Learning

This layer is used to reduce the dimensionality of map or numbers of parameter but also retains the important information. There are three pooling methods. We have used Max Pooling with matrix size of (2,2). Max Pooling selects the maximum of value in a (2 x 2) matrix.



3) Fully Connected (FC) Layer or Dense Layer - Part of Classification Layer

Before this layer the matrix is flattened into vector. This vector is feed into the FC layer to create a model with activation function to output the classification result. In this layer all the features are aggregated, and the final output is generated. We have used SoftMax activation function at output. The reason behind using SoftMax is, the CNN can give any value in the form of output. And so, all the values are scaled in the range of 0 and 1.

In CNN we had used 3 convolutional layers, 2 layers for sampling and 2 dense layers with one layer as output with output size 10.

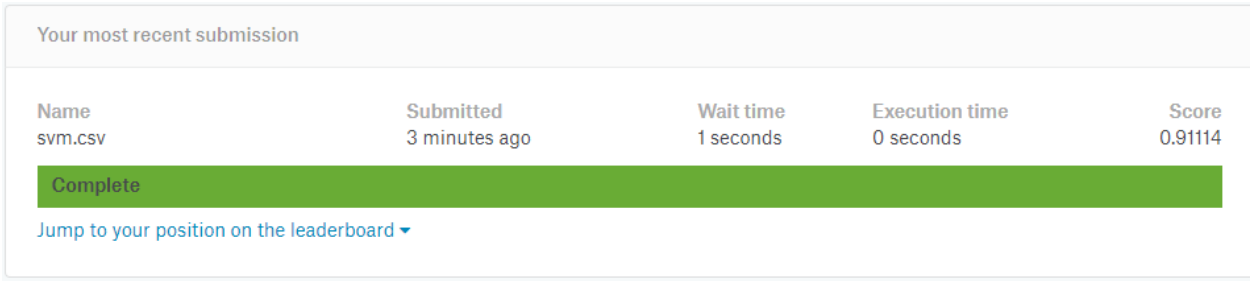
EXPERIMENTAL RESULTS AND ANALYSIS

The problem is a classic example of classification algorithm and hence we tried a few classification algorithms to see how they perform with the dataset. We used the following classification algorithms:

SVM

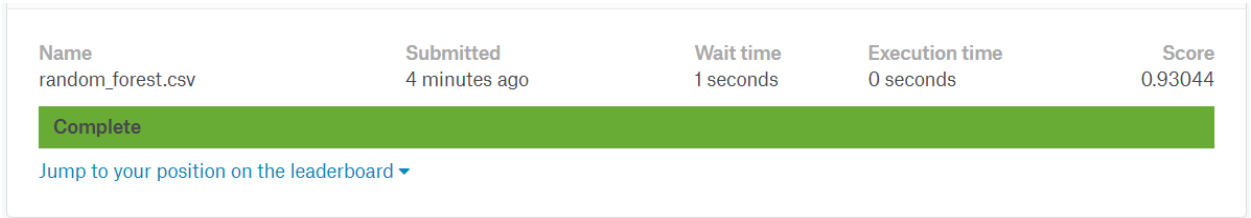
Linear SVMs try to find a hyperplane that separates the training data into binary classes with maximum margin called the margin of classification. If the data is not linearly separable, we can transform the data to make it linear. In the image of 28X28 = 784 pixels, each has its own dimension and the hyperplane divides it into two classes. The classes are from 0 to 9 and in SVM, the solution is whether a digit is correct or not, for example if a digit is 5 or not. We got an accuracy of

0.911 with the SVM. We used sklearn.svm for this experiment. Below is the image of the result as submitted in the Kaggle competition.



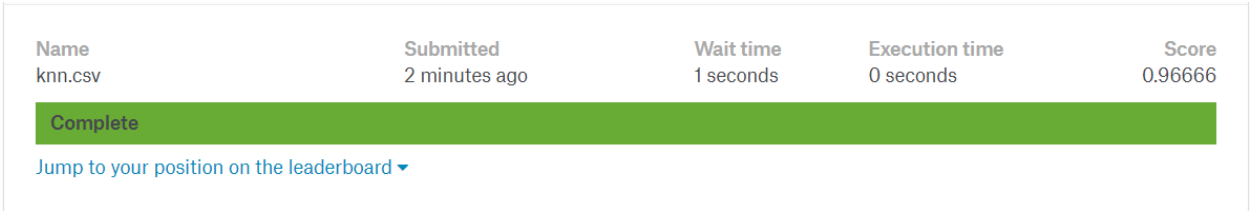
Random Forest Classifier

In the random forest algorithm, the base classifiers are decision trees. We create the decision tree models from the bootstrap samples, but each decision tree limits the splitting criteria to a random subsample of the attributes. We used the RandomForestClassifier from the sklearn.ensemble. We got an accuracy of 0.93044 and the screenshot for our submission at Kaggle is shown below.



k-Nearest Neighbor Classifier

The k-nearest neighbor classifier is the simplest image classification algorithm. The k-NN classifier was trained on the raw pixel values of the images. We used the KNeighborsClassifier from the sklearn.neighbors. We also tried running for various values of k and found that at k=5, he accuracy was highest at 0.96666. The screenshot for our submission at Kaggle is shown below.



Convolutional Neural Network

For CNN we were getting following results:

Training loss: 0.04238285419330818

Training accuracy: 0.9906349206349206

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
Output.txt	2 days ago	1 seconds	0 seconds	0.99057

Complete

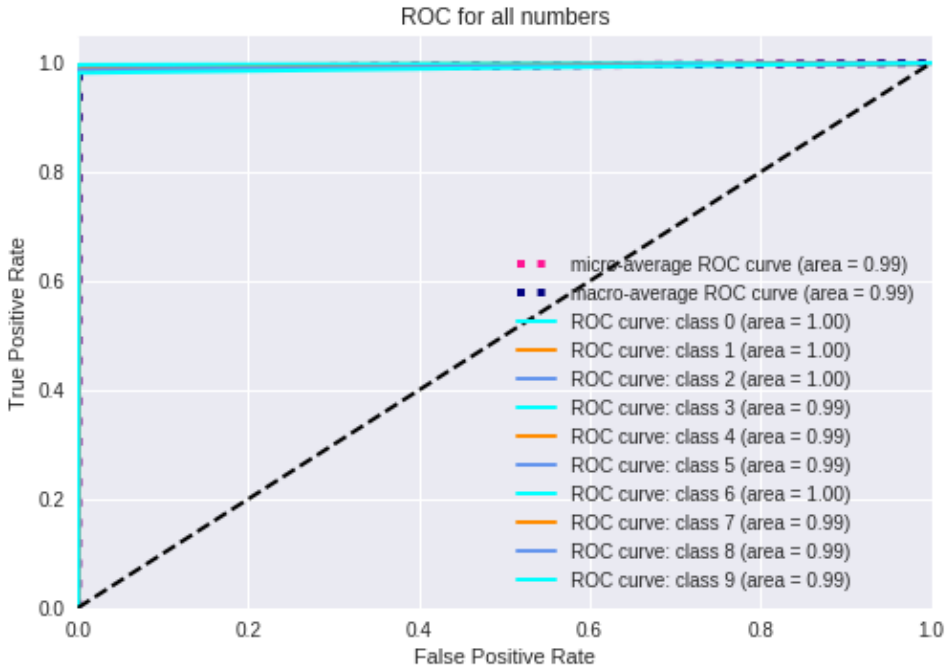
Jump to your position on the leaderboard

Recall Score = $\frac{tp}{(tp + fn)}$ where tp = number of true positives fn = number of false negatives.

Recall Score: 0.9884126984126984

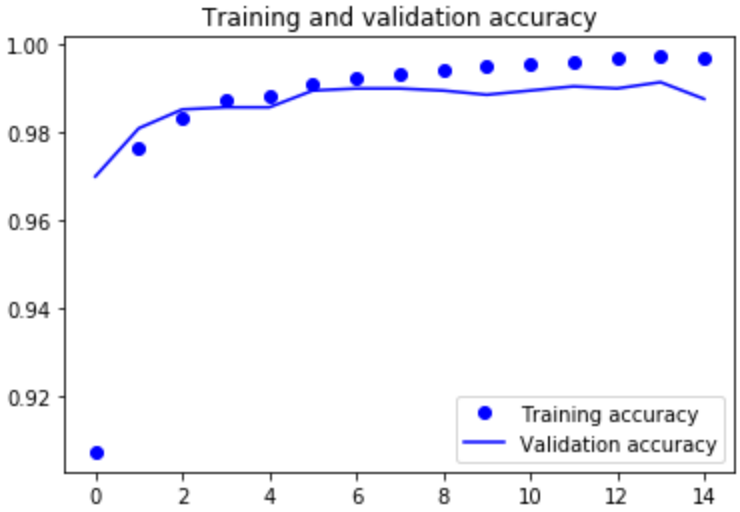
Below are the graph plots of ROC, accuracy, loss and confusion matrix.

1) ROC for all predicted numbers:

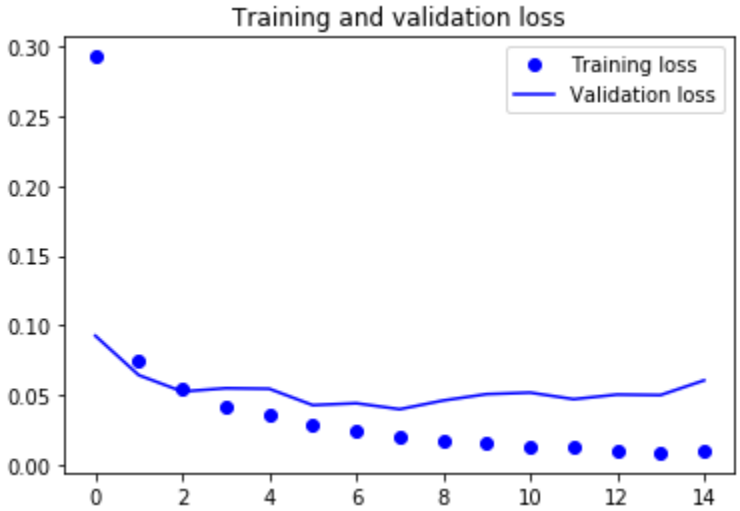


4 Numbers (0,1,2,6) are predicted with a accuracy of 100% and other numbers are predicted with accuracy of 99%.

2) Overall Training and validation accuracy plotted in one graph



3) Overall Training and validation loss plotted in one graph



4) Screen Shot of Confusion matrix (what numbers are classified what):

```
[625,   0,   1,   0,   0,   0,   1,   0,   1,   0],
[  0, 699,   0,   0,   0,   0,   1,   2,   0,   0],
[  1,   1, 640,   0,   0,   0,   0,   2,   0,   0],
[  0,   0,   2, 644,   0,   2,   0,   2,   0,   1],
[  1,   5,   0,   0, 630,   0,   1,   0,   0,   0],
[  0,   0,   0,   3,   0, 538,   3,   1,   0,   0],
[  0,   0,   0,   0,   2,   0, 631,   0,   0,   0],
[  0,   2,   3,   0,   0,   1,   0, 684,   0,   2],
[  0,   2,   1,   0,   0,   0,   4,   0, 560,   0],
[  2,   0,   0,   1,   3,   2,   0,   2,   1, 590]]
```

This image displays the confusion matrix which depicts what number in image is depicted as what. As shown 6 is classified more accurately predicted than any other number with only 2 misclassified examples out of 633. And 9 being the least accurate compared to other has 11 misclassified examples out of 601. The reason behind this can be number 9 having similarity in curve with 8, 0, 5, 3.

CONCLUSION

Accuracy comparison of Kaggle:

Classifier	Accuracy
Convolutional Neural Network	0.99057
Support Vector Machine	0.91114
k-Nearest Neighbor	0. 96666
Random Forest	0. 93044

From above statistics CNN performs the best among all.

We are getting the Area under ROC curve as 0.9934913309862404.

Four labels (0,1,2,6) are having Area under ROC curve as 1. And other are having 0.99.

It is because CNN being a deep learning classifier, learns all features in the form of weight matrix in Convolution layer.

The accuracy can still be increased by using all the training data for learning. And to further increase the accuracy one can implement reinforcement learning i.e. give the model an image if it classifies incorrectly then add the image to the dataset and then correctly label it, in this way the model will have more specific knowledge about each feature of a digit.

CONTRIBUTION OF TEAM MEMBERS

Project Selection and Brainstorming – Debanjana, Paril, Vivek
Research on the relevant tools and technologies – Paril, Vivek
Related work - Debanjana
Dataset Cleaning - Paril, Vivek
Dataset Preprocessing - Debanjana, Vivek
SVM classification - Debanjana
Random Forest classification – Paril
K-NN classification - Vivek
CNN classification - Debanjana, Paril, Vivek
Evaluation Metrices - Debanjana, Paril, Vivek
Analysis of results - Debanjana, Vivek
Conclusion - Paril

REFERENCES

1. https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#sphx-glr-auto-examples-model-selection-plot-roc-py
2. https://www.google.com/urlsa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiL_MTokv3eAhVQd6wKHalTD14QjRx6BAgBEAU&url=https%3A%2F%2Fwww.tensorflow.org%2Fguide%2Ffeature_columns&psig=AOvVaw1R508yyvHBauslhSPaXvFt&ust=1543702559262544
3. Images and text from class slides
4. Paper - “A Few Useful Things to Know about Machine Learning” by Pedro Domingos.
5. https://keras.io/layers/convolutional/