STEEL SURFACE DEFECT MACHINE LEARNING TERM PROJECT AUGUST 21, 2019

BASED ON THE NORTHEASTERN UNIVERSITY (NEU) DEFECT DATABASE

GROUP MEMBERS:

MASTHANAIAH PELLURI (MAST311@GMAIL.COM)
NARESH PATEL (NP.PATEL@OUTLOOK.COM)
KERIM TERZIOGLU (KTERZIOGLU@YAHOO.COM)
JITENDRAKUMAR PRAJAPATI (JEETPRAJAPATI@GMAIL.COM)

Content

1. Project Overview
2. Dataset Overview
3. Data Analysis / Feature Engineering
4. Models
5. Deep Learning using Convolutional Neural Network
6. Model Evaluation
7. Challenges
8. Conclusion

Project Overview

Steel is the most important building materials of modern days. Surface quality of steel is essential for steel industry and detecting quality issue is very challenging.

Goals and Objective :

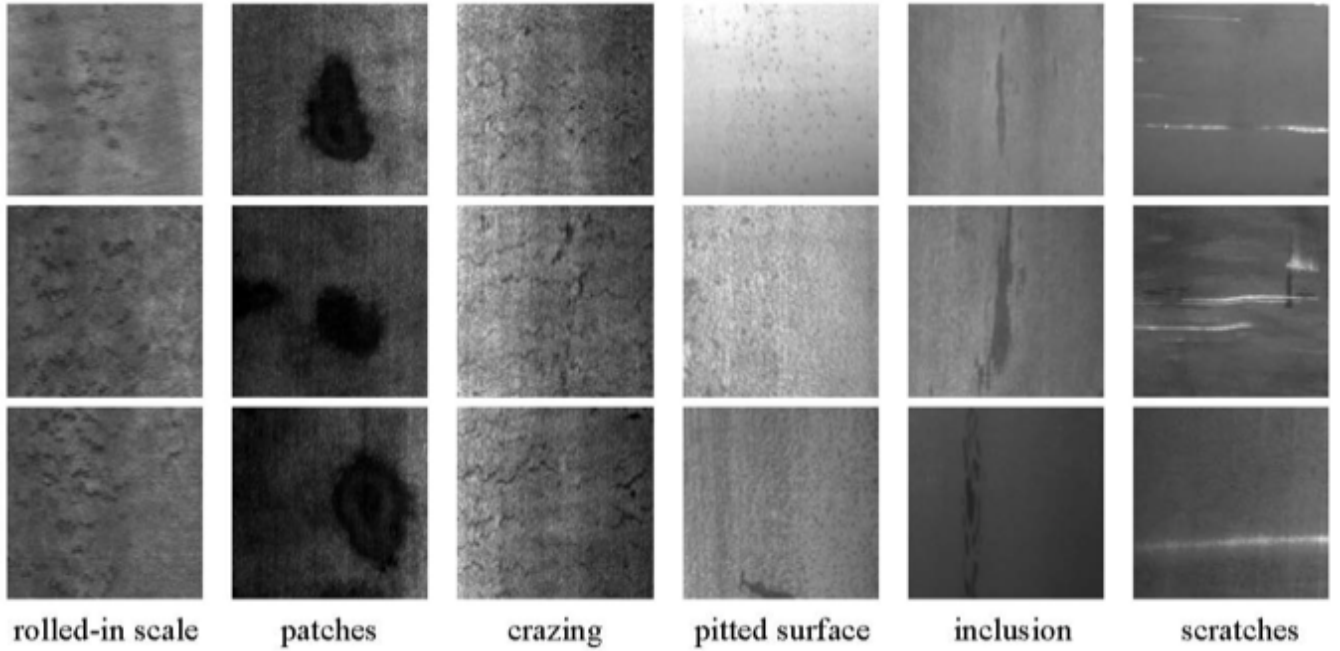
The challenge is to detect and classify steel surface defects using machine and deep learning. Accuracy metrics is used to evaluate the models.

Accuracy = Total Number of Correct Predictions / Total number of Images

Dataset overview

In the NEU surface defect database, six kinds of typical surface defects of the hot-rolled steel strip are categorized:

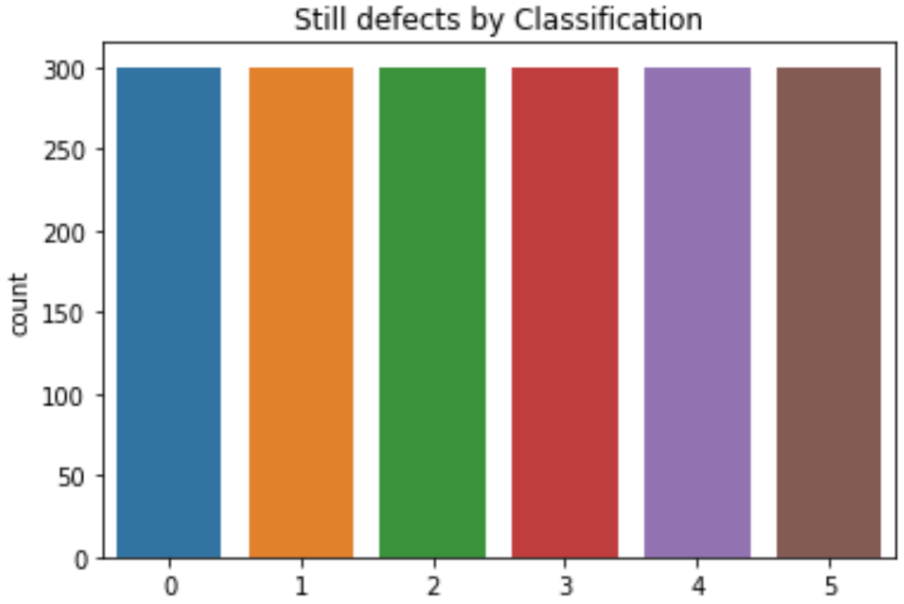
1. RS - rolled-in scale
2. PA - patches
3. CR - crazing
4. PS - pitted surface
5. IN - inclusion
6. SC - scratches



The database includes 1,800 grayscale images: 300 samples each of the six typical surface defects categorized above.
http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

Data Analysis / Feature Engineering

In the Northeastern University (NEU) surface defect database, six kinds of typical surface defects of the hot-rolled steel strip are collected, i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). The database includes 1,800 grayscale images: 300 samples each of six different kinds of typical surface defects. Dataset is well balanced as we can see in below image



0 - Crazing

1 - Inclusion

2 - Patches

3 - Pitted surface

4 - Rolled-in-scale

5 - Scratches

Data Analysis / Feature Engineering

Utilized 2 scikit-image.org APIs:

1. greycomatrix - calculate the grey-level co-occurrence matrix (GLCM) for a given image
   A grey level co-occurrence matrix is a histogram of co-occurring grayscale values at a given offset over an image.
2. greycoprops - calculate the texture properties of a GLCM

- 'contrast': $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$
- 'dissimilarity': $\sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$
- 'homogeneity': $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- 'ASM': $\sum_{i,j=0}^{levels-1} P_{i,j}^2$
- 'energy': $\sqrt{ASM}$

```
image_xlx.head()
```

	0	1	2	3	4	5	6
0	NaN	contrast	dissimilarity	homogeneity	ASM	energy	Label
1	0.0	12769357	562393	2747.88	312782	559.269	0
2	1.0	9580203	482361	3308	289038	537.623	0
3	2.0	10928946	517098	3084.55	337650	581.077	0
4	3.0	12465011	556457	2776.65	372854	610.618	0

*ASM : Angular Second Moment
More info on link GLCM Book : Practical Computer Vision Applications Using Deep Learning with CNNs

Data Analysis / Feature Engineering

Sources https://link.springer.com/chapter/10.1007/978-981-10-7242-0_7
http://cpb.iphy.ac.cn/article/2017/1901/cpb_26_9_098104.html#

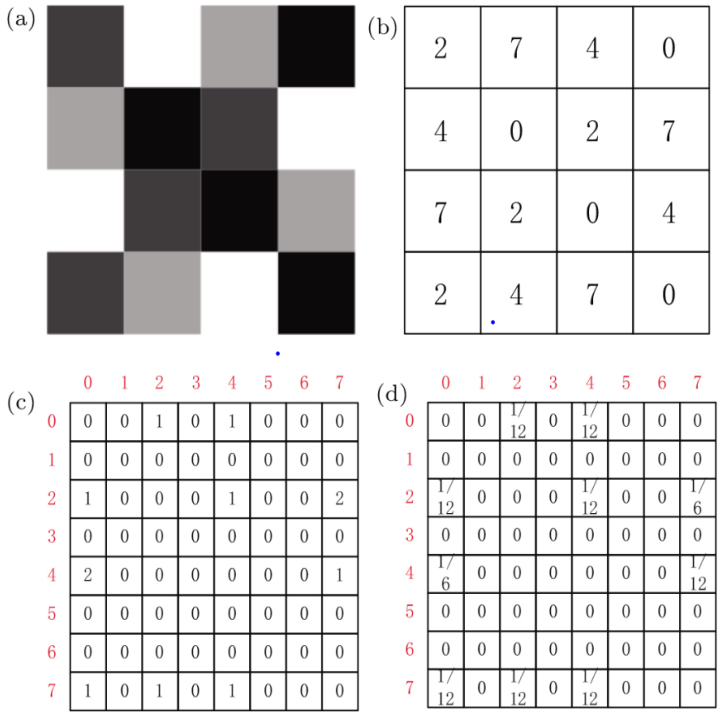
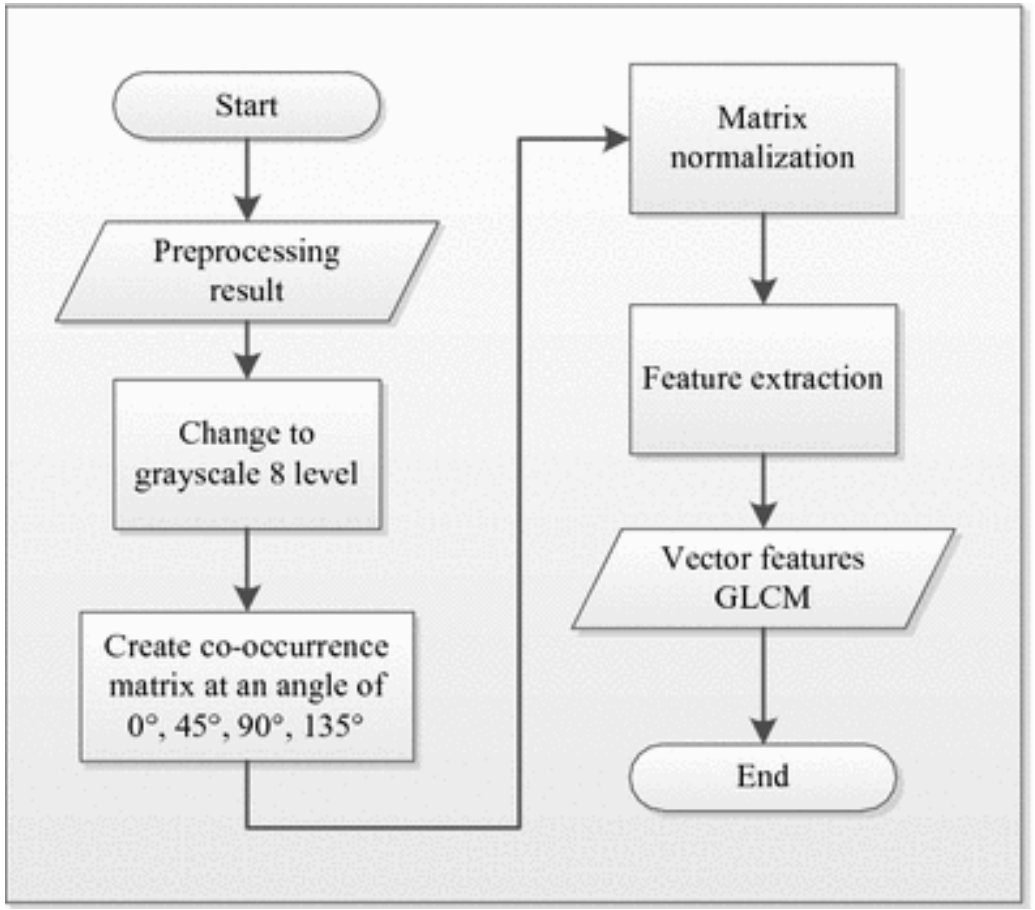


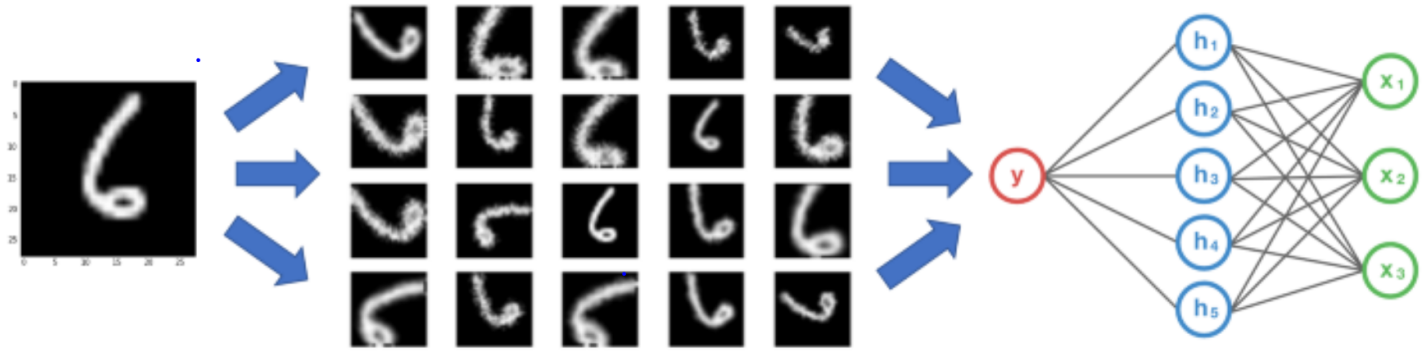
Fig. 1. (color online) An example to show how to obtain GLCM from an image. (a) Original image *f*, (b) matrix of image *f*, (c) initial GLCM (d = 1, θ = 0°), and (d) normalized GLCM (d = 1, θ = 0°).



Data Analysis / Feature Engineering

Data Augmentation :
To train convolutional neural network, we have used data augmentation strategy.

Data augmentation is strategy that is used for increasing the size of a training dataset by creating modified images without collecting new data. Example of data augmentation techniques : Rotating, Cropping, padding and flipping(horizontally or vertically) the images



Data Augmentation in play

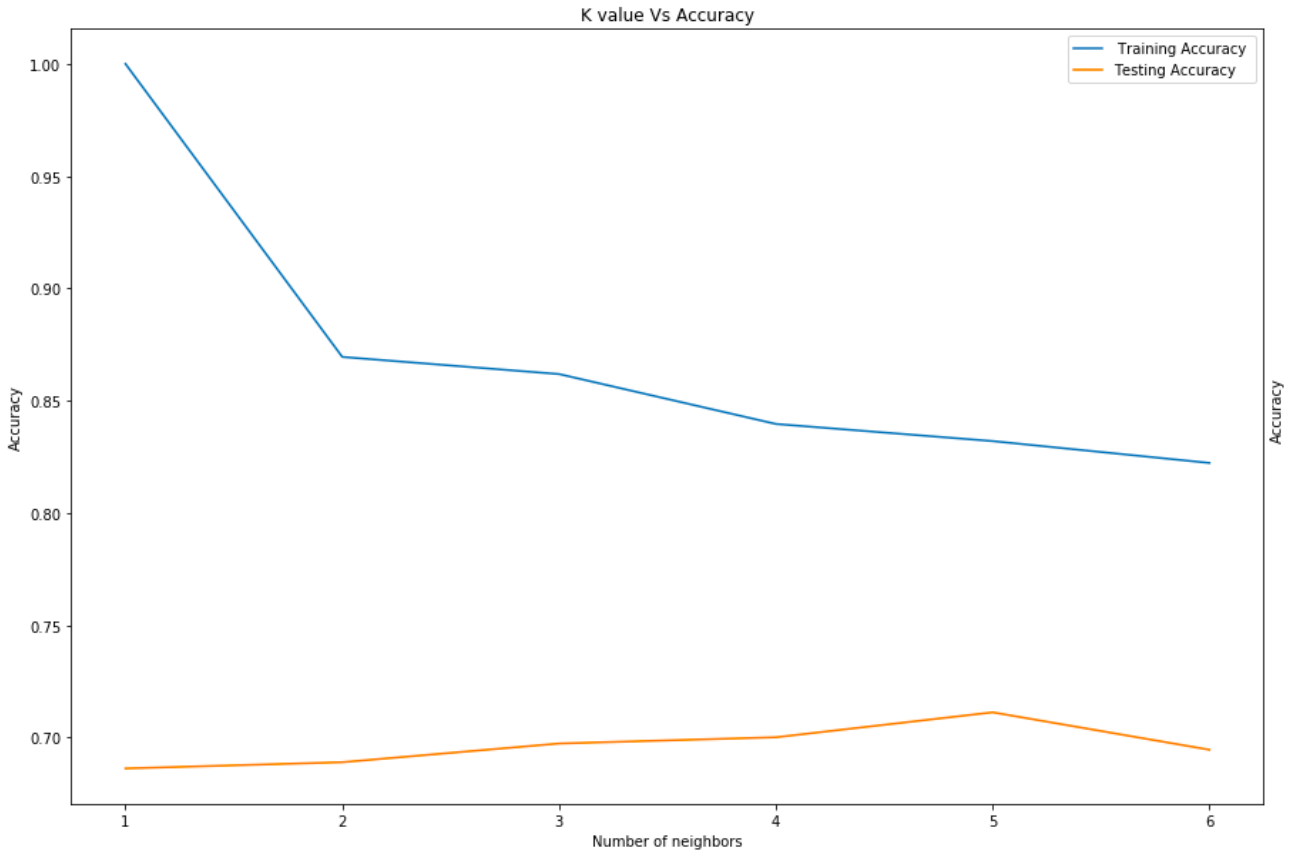
Sources : https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/
https://bair.berkeley.edu/blog/2019/06/07/data_aug/
https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

Models

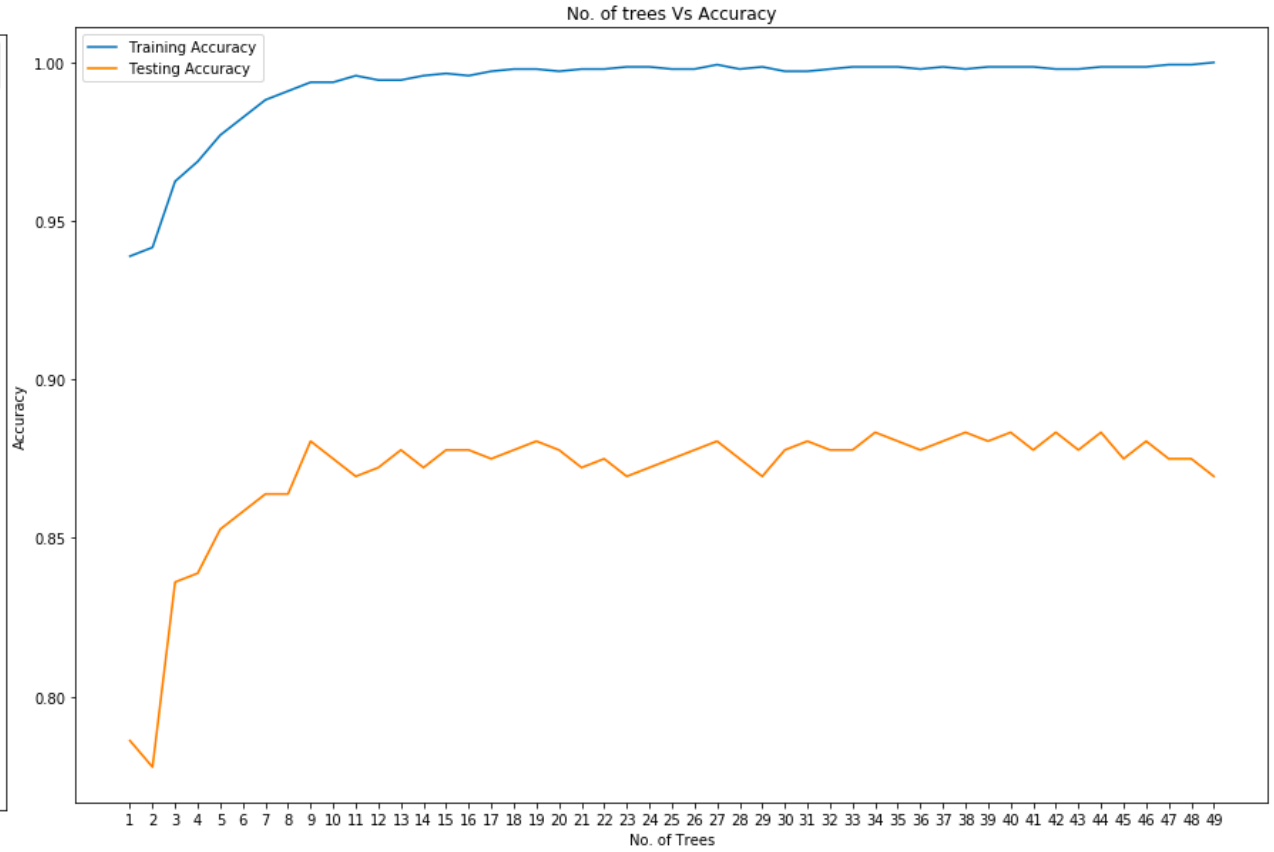
KNN
Random Forest
Decision Tree with Boosting
SVM with Grid Search
CNN with Data Augmentation
CNN without Data Augmentation

Model Evaluation

KNN: Varying Number of Neighbors, K=5

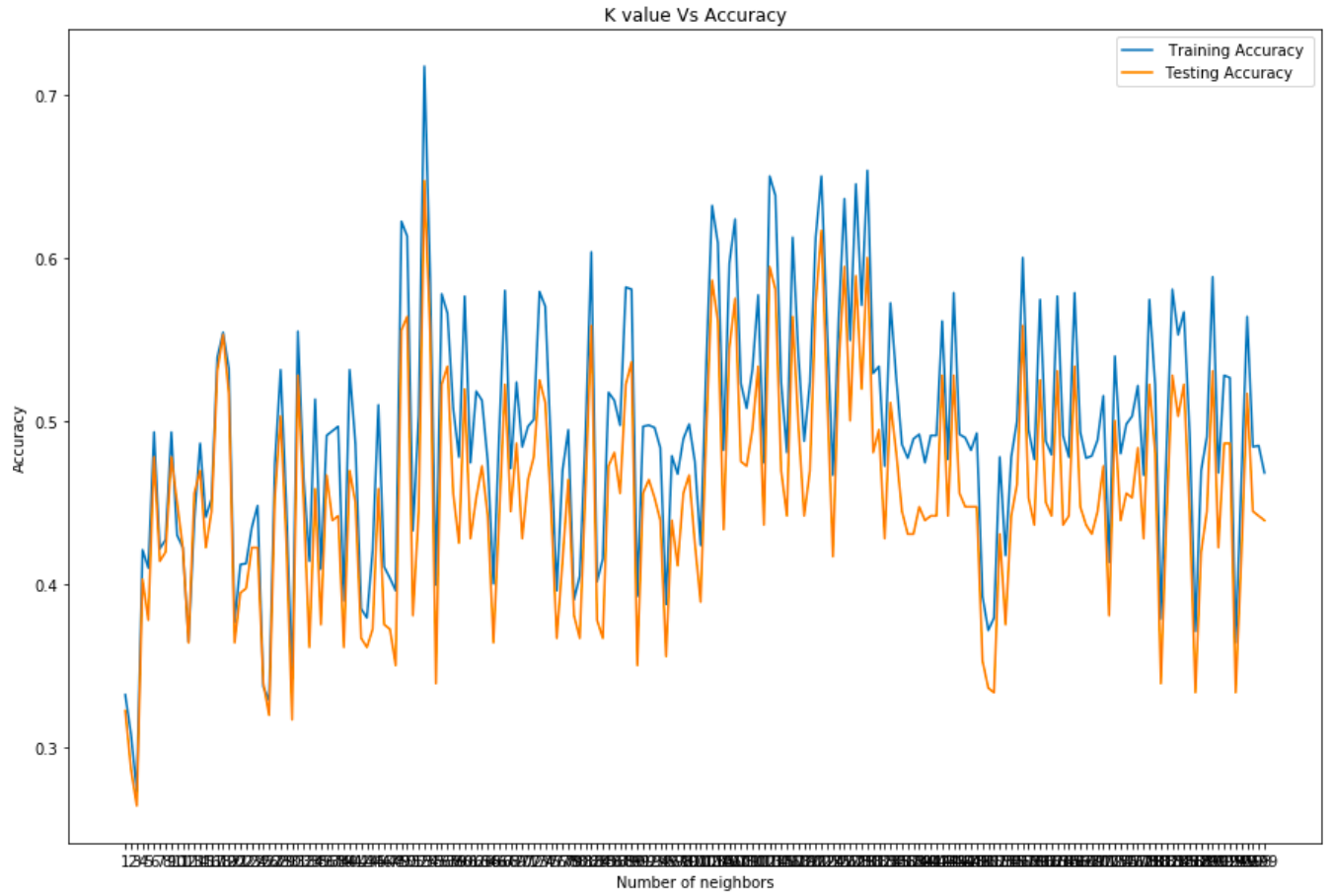


Random Forest: Number of Trees = 34



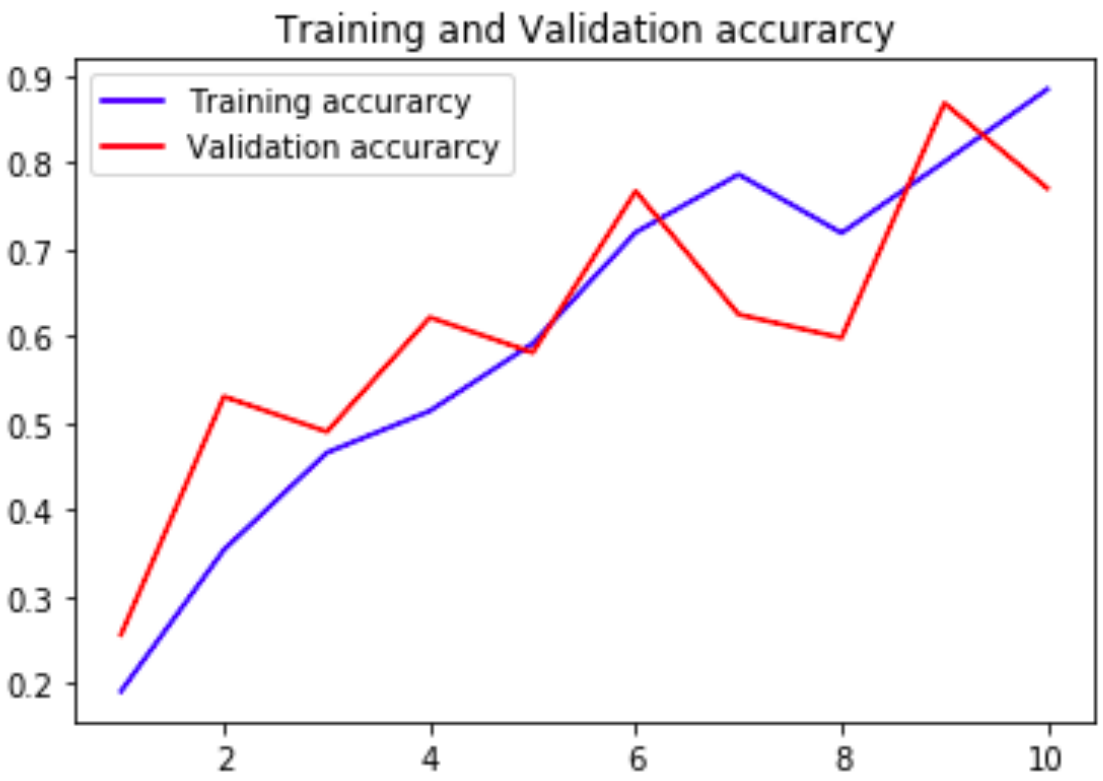
Models Evaluation

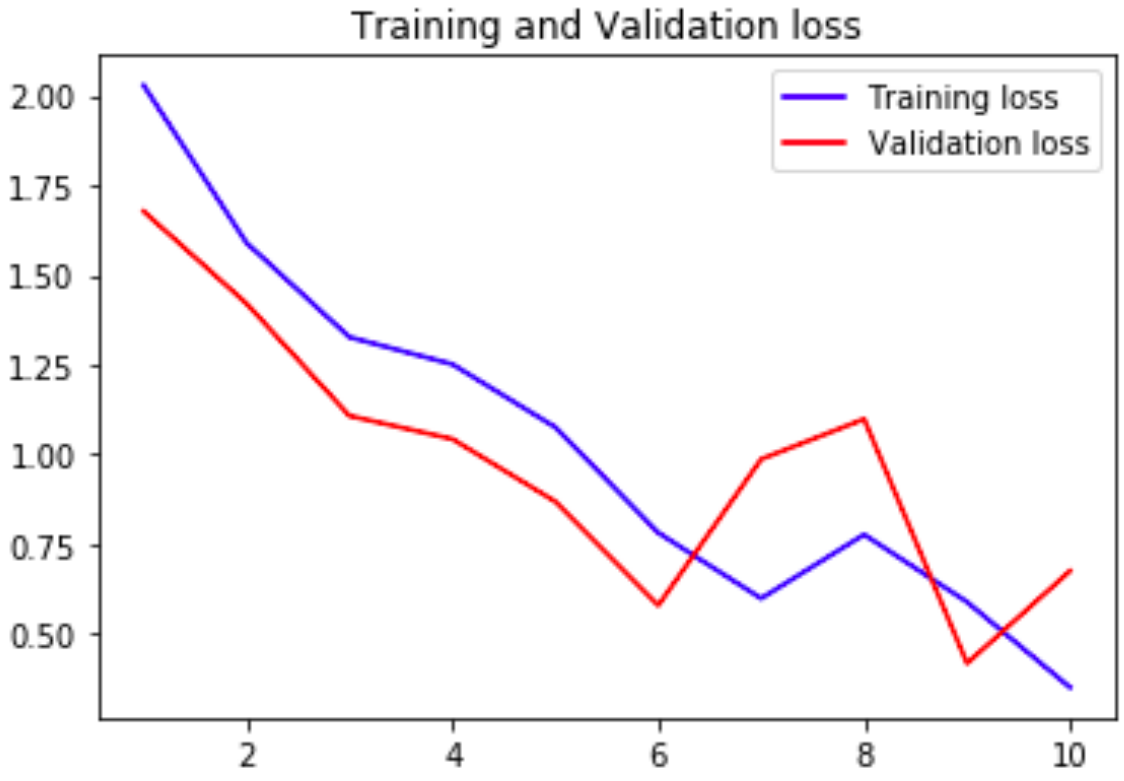
Decision Tree with Boosting: K=53



Deep Learning using Convolutional Neural Network

CNN with Data Augmentation





Models Evaluation

Models	ACCURACY	PARAMETERS
KNN	71.1%	K=5
Random Forest	88.3%	Number of Trees=34
Decision Tree with Boosting	64.7%	K=53
SVM with Grid Search	0.17%	C': 0.001, 'gamma': 0.001
CNN with Data Augmentation*	75.55%	Conv2D, Dropout (2 layers each)
CNN without Data Augmentation	88.06%	Conv2D, Dropout (2 layers each)

Data Augmentation was introduced for the Neural Network.

Challenges

1. Grayscale Images
2. Extracting Features of images
3. Models performance due to Grayscale Images converted to 3 channels
4. Models performance after Image augmentation

Conclusion

- Based on our research and after utilizing the techniques learned during the class, our conclusion is that the Random Forest model has the best accuracy score and hence can be treated as a Best Model for this dataset
- The features were extracted using GLCM
- Learned how to use CNN for smaller dataset

- Future Usage and Enhancements:
  - Use images process for Kaggle Competition
  - Use different technique to get better performance of the models