MANUFACTURING DEFECT RECOGNITION

Course Project
MACHINE LEARNING - CSP 774

Pranav Reddy
2017ume0110

CONTENTS

OVERVIEW

Brief about the paper & my contributions

THE PAPER

- The paper is titled
  "One shot recognition of manufacturing defects in steel surfaces"
- The paper proposes the application of siamese convolutional neural network to do one shot recognition of manufacturing defects
- To show the effectiveness of the model, the results are compared with
  CNN (of same architecture)
  KNN (K nearest neighbours)

MY OBSERVATIONS

- The model had been trained directly on images with minimal preprocessing of the data
- Images have been converted to histograms and directly used on the model
- Many other traditional machine learning models, which are computationally less expensive could give better results. Exhaustive comparison hasn't been made

BACKGROUND

Need for research - what & why?

INSPIRATION

- AI, Machine intelligence is the need of the hour in manufacturing sector
- Tonnes of material, labour, energy & REVENUE is lost when defected workpieces are manufactured / unrecognized
- Deep learning + computer vision made it possible with near human accuracy also providing the advantage of non invasive inspection

WHY SIAMESE?

- Smart manufacturing is a data intensive approach. Large amounts of data needs to collected, stored & labelled.
- Deployment of traditional ML/ DL models is either limited by accuracy or time consuming & computationally expensive
- WIth siamese CNN, one shot recognition is possible, drastically reducing the training data requirements. This would mean that the model could also be run in real time
- If a new product is manufactured, then new training data would be required.

THE ALGORITHM

- The key idea behind the one shot recognition is that the model should recognize the class correctly with only a single sample
- During training, it learns to identify the differences in features of the input image pair
- The model would then return the degree of similarity between any random image pair, even if the data is unique
- By setting a threshold to this degree, the given image can be classified as that of the same class of its counterpart or different

DATA SET

Source & type of data

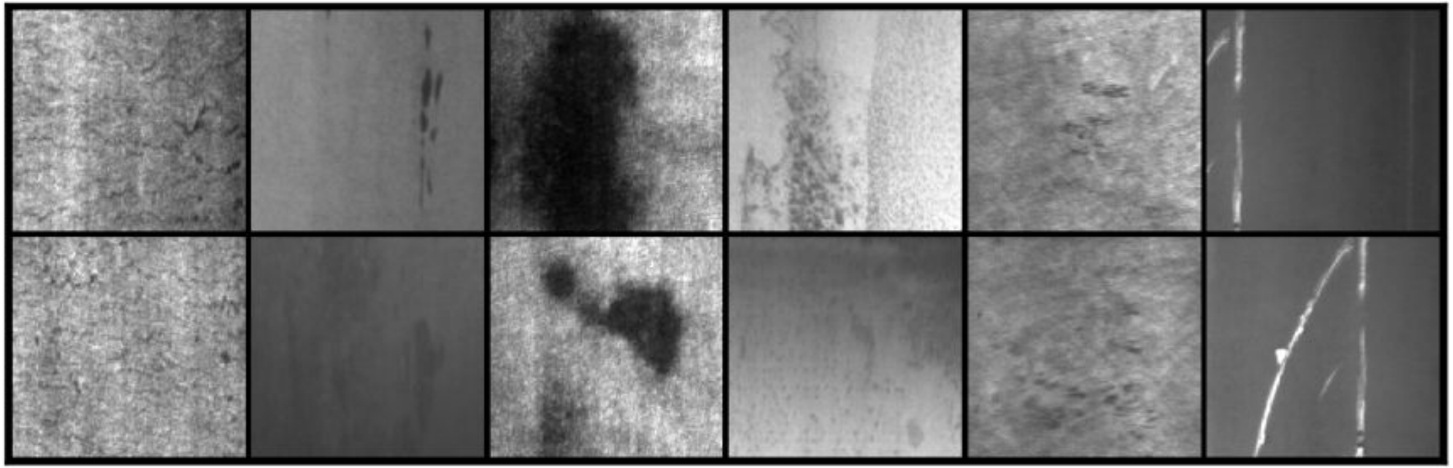
Link for data set :

http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

The data set is from 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. the original resolution of each image is 200×200 pixels.



Random samples for each class in columns from left to right: crazing (Cr), inclusion (In), patches (Pa), pitted surface (PS), rolled-in scale (RS) and scratches (Sc)

The dataset images have a variation in illuminations which introduces further challenges for the recognition. This variability results in large differences in samples belonging to the same class. Another challenge that can be observed is the similarity in images belonging to different classes. For example, the similarity in images belonging to the categories of crazing and rolled-in scale steel surfaces is easily noticeable.

PRE PROCESSING

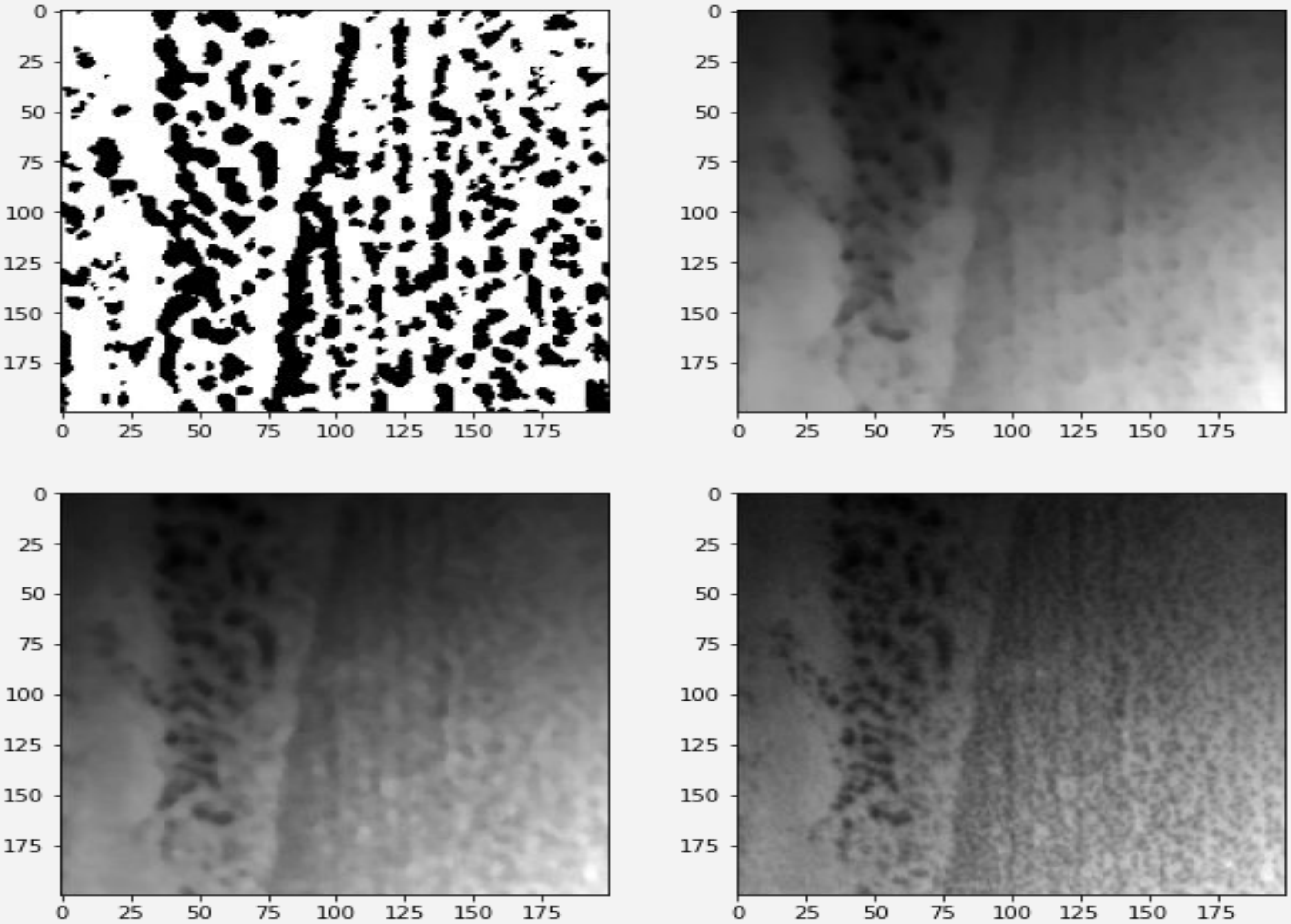
Image processing for better recognition

IMAGE PROCESSING

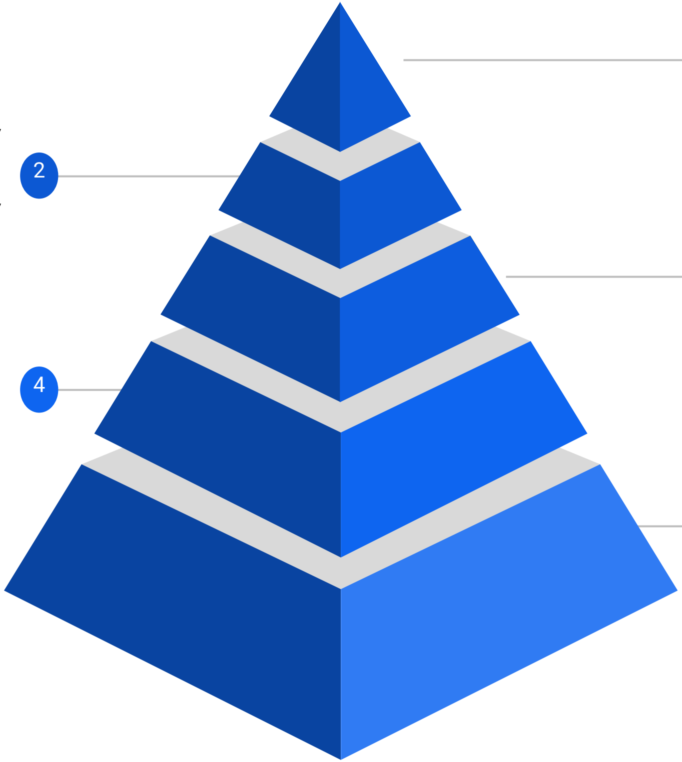
- The performance of the model depends heavily on feature extraction and its pre processing especially when data is images
- For better recognition of defects the images need to be scale invariant, noise free, equalised brightness
- The model would then return the degree of similarity between any random image pair, even if the data is unique
- By setting a threshold to this degree, the given image can be classified as that of the same class of its counterpart or different

PRE PROCESSING EXP 1

MEDIAN BLUR	OPENING	THRESHOLDING
Almost all images are noise-rich. To reduce noise, uneven brightness concentration, median blur filter has been applied. OpenCV median blur gave better results than Scipy's	Structuring elements which can open the image in 6 directions (angles) have been used. CV2's opening filter applied for each element. 6 images are added together	CV2's adaptive gaussian mean thresholding has been applied to retain only the most important features. The image has been closed by using elliptical str. Element for better features


0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175
0 25 50 75 100 125 150 175

PRE PROCESSING EXP 2



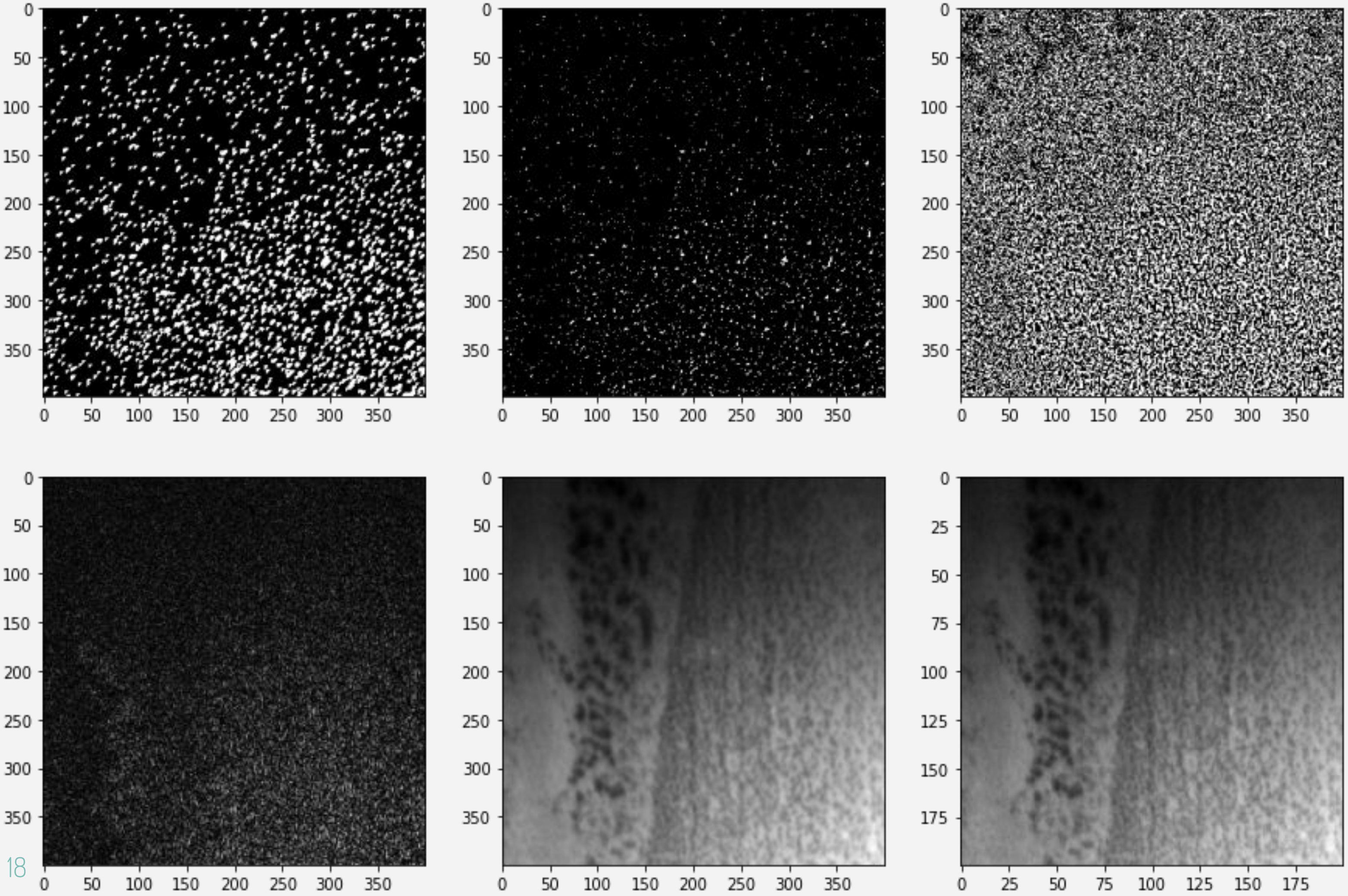
1. **Gaussian blur**
To remove excess noise. To smoothen the image & reduce uneven brightness with gaussian mask

2. **Laplacian**
To retain high frequency components like edges and corners. Used from openCV

3. **Threshold**
To preserve the original structure. Used adaptive thresholding from openCV

4. **Erode**
Done as part of closing image as huge amount of noise is present. Used elliptical structuring element for 2 iterations

5. **Dilate**
To highlight main features, by increasing its size. Used elliptical structural element for 3 iterations



PRE PROCESSING EXP 3

Intensity re scaling

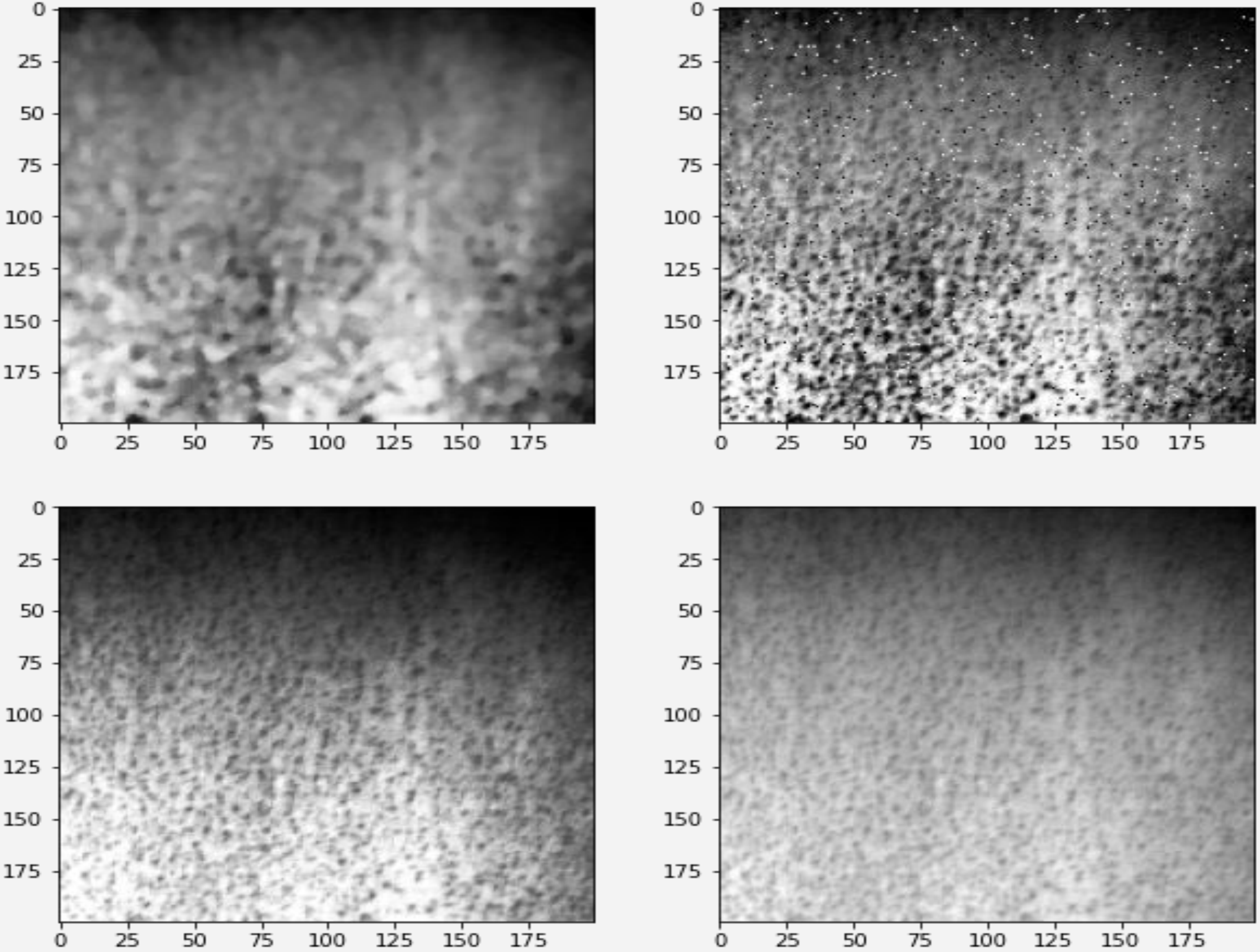
Skimage's exposure module used. Intensity values of image are shrinked or stretched according to desired input values

Histogram equalization

Skimage's exposure module. Contrast of image adjusted using histogram equalization. Uneven brightness problem is solved

Median filter

Histogram equalization results in large amount of noise. Median filter applied to reduce it. Low frequency components are lost


0
25
50
75
100
125
150
175
0 25 50 75 100 125 150 175

FEATURE EXTRACTION

Extracting appropriate features

FEATURE VECTOR

- 2 Types of feature vectors have been created for each pre processing experiment

1) An n*n Image directly flattened into an n^2*1 array
2) The histogram of the image is calculated. It's values are then normalized and converted into an array

- The models, thus would be trained on 6 types of data
- The CNN model has directly been trained on images, as it generates feature vectors on its own

MIN MAX SCALING

- Many machine learning algorithms work better when features are on a relatively similar scale and close to normally distributed
- Transforms features by scaling each feature to a given range.
- This estimator scales and translates each feature individually such that it is in the given range on the training set, (Default - between 1 & 0)
- Could be used for logistic regression, nearest neighbors, neural networks, support vector machines

MODEL SELECTION

Comparison between models

SIAMESE CNN

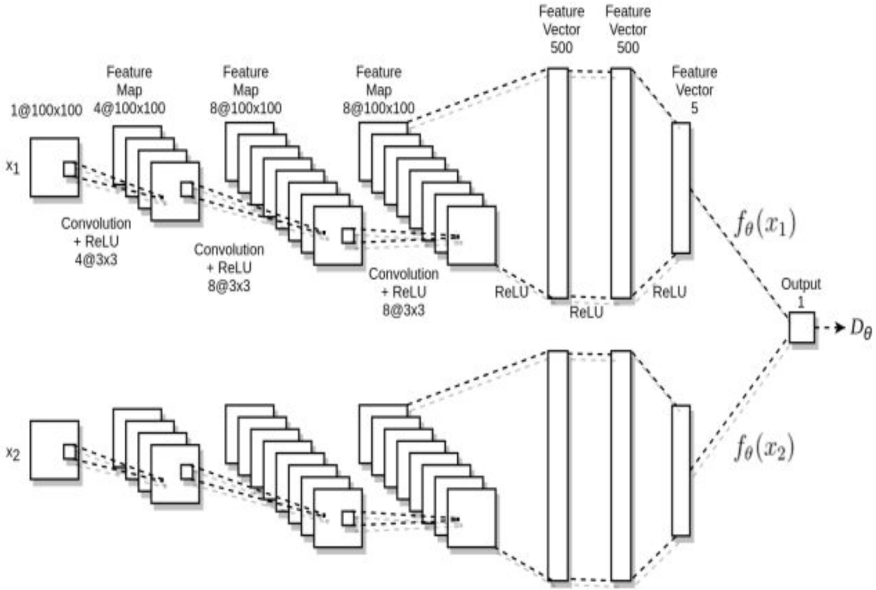
- The model of primary interest is siamese CNN.
- Deep learning models like CNN do not demand of feature engineering.
- Siamese CNN also provides the advantage of training the system on one set of defects and testing on completely different set. This is not possible in a traditional classification cycle.

CONVENTIONAL ML MODELS

- Though deep learning algorithms outweigh traditional ML algorithms in many ways, They are still computationally very expensive and time consuming to deploy
- Hence the ML models have also been chosen for an exhaustive comparison of performance with siamese CNN and to see whether they give a better trade off between computational requirement and accuracy

MACHINE LEARNING MODEL	PARAMETERS
K nearest neighbours	No of neighbours = 1, uniform weights, no of jobs run in parallel = 1
Support vector machine	Regularization parameter C= 3.2
Support vector machine (Kernel)	Regularization parameter C= 4, Kernel = rbf
Logistic Regression	Float (inverse regularization strength) = 1, solver = lbfgs
Decision trees	Min samples to split = 2, Max feature cap = none
Gaussian Naive Bayes	Prior probabilities of class = None
Random forest	No. of estimators = 100, Min sample split = 2, Max depth = None
Neural networks	2 layers = (62,62) neurons, solver = lbfgs, activation = tanh

MODEL PARAMETERS – SIAMESE CNN




```
(conv_layer): Sequential(
  (0): ZeroPad2d(padding=(1, 1, 1, 1), value=0.0)
  (1): Conv2d(1, 4, kernel_size=(3, 3), stride=(1, 1))
  (2): ReLU()
  (3): ZeroPad2d(padding=(1, 1, 1, 1), value=0.0)
  (4): Conv2d(4, 8, kernel_size=(3, 3), stride=(1, 1))
  (5): ReLU()
  (6): ZeroPad2d(padding=(1, 1, 1, 1), value=0.0)
  (7): Conv2d(8, 8, kernel_size=(3, 3), stride=(1, 1))
  (8): ReLU()
)
(fc_layer): Sequential(
  (0): Linear(in_features=80000, out_features=500, bias=True)
  (1): ReLU()
  (2): Linear(in_features=500, out_features=500, bias=True)
  (3): ReLU()
  (4): Linear(in_features=500, out_features=5, bias=True)
)
```

CONTRASTIVE LOSS FUNCTION

For training, contrastive loss function has been used.

The loss function is parameterized by the weights of the neural network θ and the training sample i. The i th training sample from the dataset is a tuple (x1, x2, y) where x1 and x2 are pair of images and the label y is equal to 1 if x1 and x2 belong to same class and 0 otherwise

$$L(\theta, (x_1, x_2, y)^i) = y\frac{1}{2}D_{\theta,i}^2 + (1-y)\frac{1}{2}(max\{0, m - D_{\theta,i}\})^2$$

PARAMETERS	VALUES
Batch size	31
No of epochs	80
Learning rate	0.0005
Neural network optimizer	Adam
Adam parameters (β1, β2)	0.9, 0.999

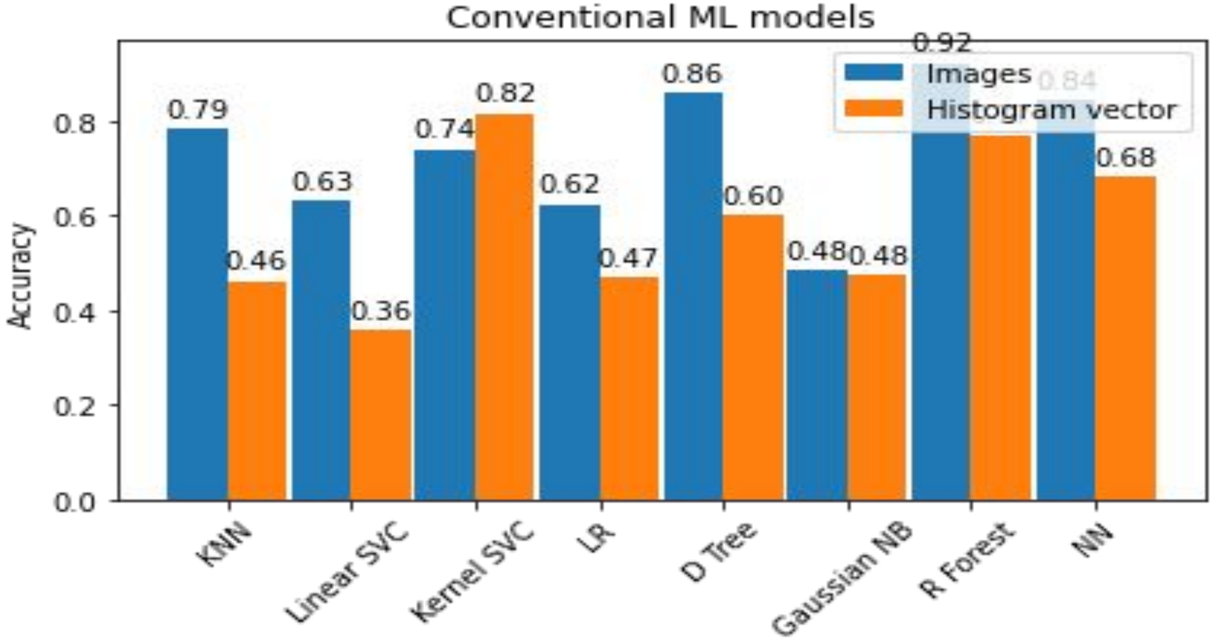
- The dataset was divided into two sets. The training set consisted of 3 classes - rolled in scale, patches, inclusion. crazing, pitted-surface and scratches were shown to the network in the testing phase for one-shot recognition.
- Data samples were chosen randomly during training. While sampling an image pair, 2 images were chosen from the same category with a probability of 0.5 with a corresponding label of y = 1. Similarly, the images were chosen from two different categories with the remaining probability of 0.5 with label y =0

RESULTS

Evaluation of the model using metrics

ACCURACY - ML (NO PP)

Accuracies of 8 models with raw image matrices, their corresponding histogram vectors



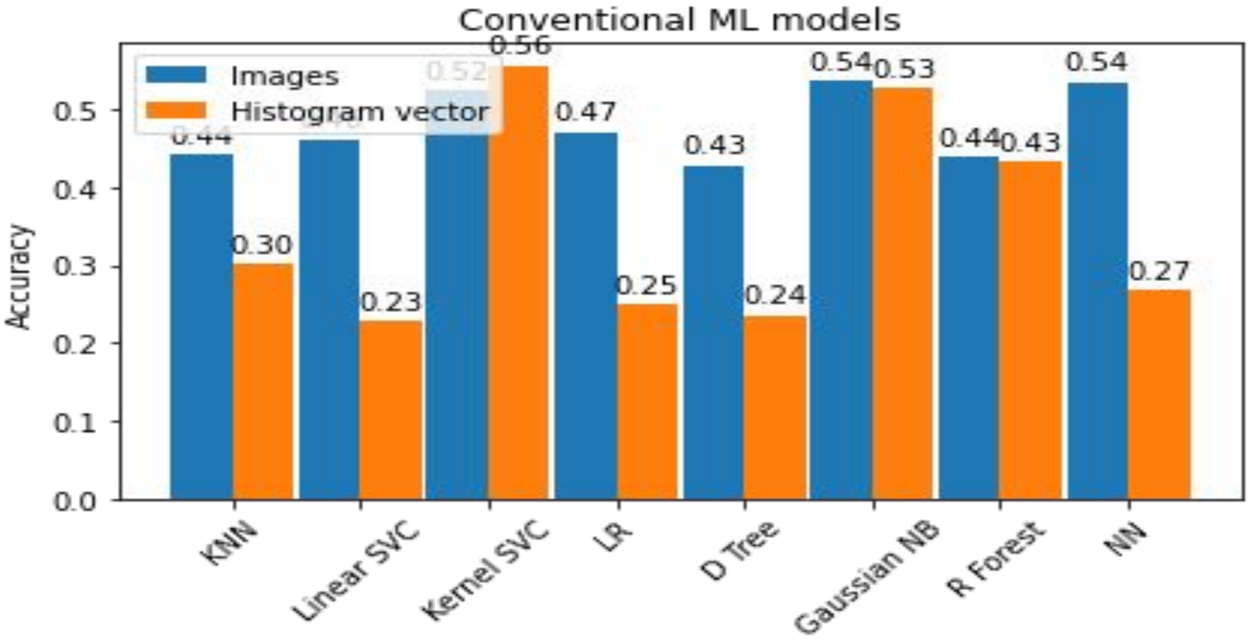
CLASSIFICATION REPORT

```
Confusion matrix for Random forrest classifier
[[74  0  2  1  1  1]
 [ 0 52  0  4  1  1]
 [ 2  0 68  0  0  0]
 [ 3  3  1 77  1  0]
 [ 0  8  0  1 56  1]
 [ 0  2  0  0  2 58]]

              precision    recall  f1-score   support

           1       0.94      0.94      0.94        79
           2       0.80      0.90      0.85        58
           3       0.96      0.97      0.96        70
           4       0.93      0.91      0.92        85
           5       0.92      0.85      0.88        66
           6       0.95      0.94      0.94        62

    accuracy                           0.92       420
   macro avg       0.92      0.92      0.91       420
weighted avg       0.92      0.92      0.92       420

Micro average is :
 0.9166666666666666
```

Random forest classifier with direct pixel values of image as feature vector has performed better than other models

ACCURACY - ML (PP - 1)

Accuracies of 8 models with raw image matrices, their corresponding histogram vectors



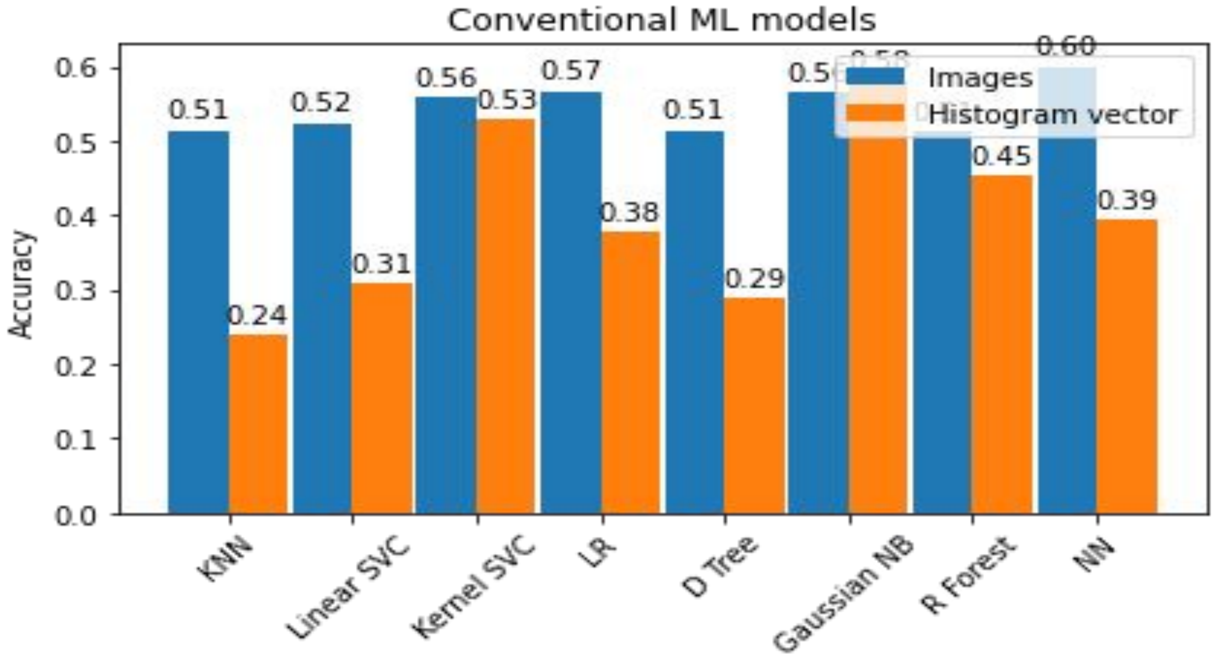
CLASSIFICATION REPORT

```
Confusion matrix for Random forrest classifier
[[55  0 18  2  4  0]
 [ 0 41  0  3  0 14]
 [24  0 39  4  3  0]
 [12  5  5 30 21 12]
 [ 2  0  1 16 46  1]
 [ 1 22  2  8  6 23]]

              precision    recall  f1-score   support

           1       0.59      0.70      0.64        79
           2       0.60      0.71      0.65        58
           3       0.60      0.56      0.58        70
           4       0.48      0.35      0.41        85
           5       0.57      0.70      0.63        66
           6       0.46      0.37      0.41        62

    accuracy                           0.56       420
   macro avg       0.55      0.56      0.55       420
weighted avg       0.55      0.56      0.55       420

Micro average is :
 0.5571428571428572
```

SVM (Kernel) Classifier with values of histogram of image as feature vector has performed better than other models

ACCURACY - ML (PP - 12)

Accuracies of 8 models with raw image matrices, their corresponding histogram vectors



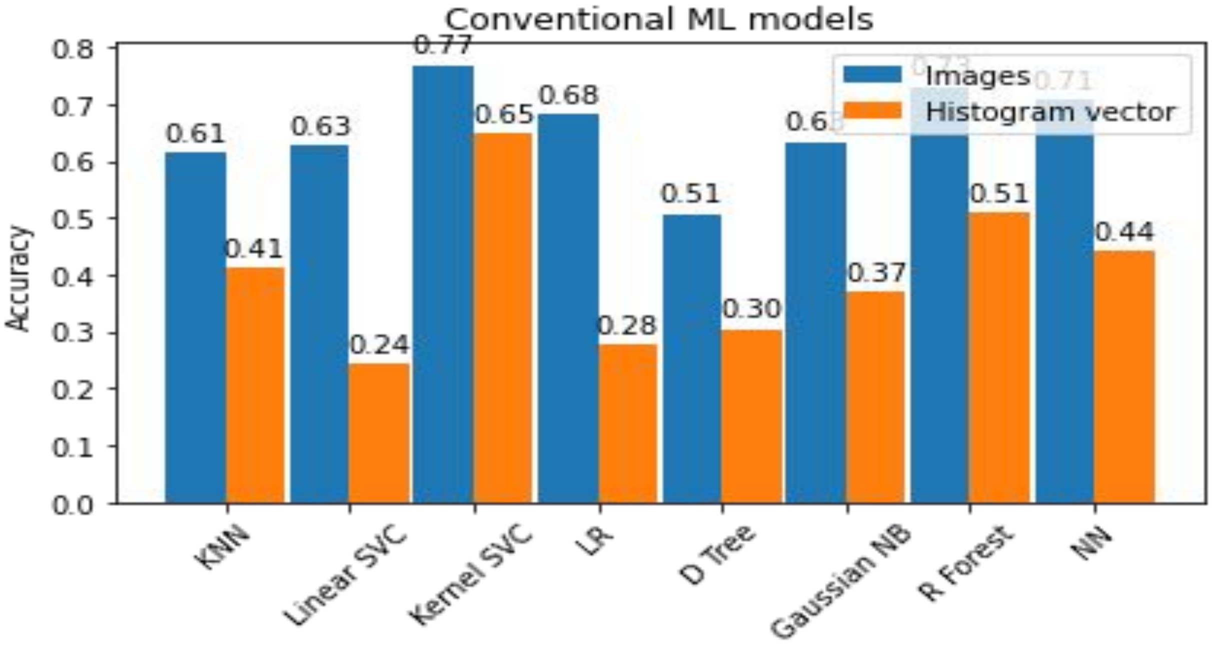
CLASSIFICATION REPORT

```
Confusion matrix for Neural net MLP Classifier (2 layers size = 62)
[[49  0 25  0  5  0]
 [ 0 46  0  0  0 12]
 [24  0 34  1 11  0]
 [ 1  4  3 20 32 25]
 [ 4  0  4  5 53  0]
 [ 0 10  0  2  0 50]]

              precision    recall  f1-score   support

           1       0.63      0.62      0.62        79
           2       0.77      0.79      0.78        58
           3       0.52      0.49      0.50        70
           4       0.71      0.24      0.35        85
           5       0.52      0.80      0.63        66
           6       0.57      0.81      0.67        62

    accuracy                           0.60       420
   macro avg       0.62      0.62      0.59       420
weighted avg       0.62      0.60      0.58       420

Micro average is :
 0.6
```

Neural Network Classifier with direct pixel values of image as feature vector has performed better than other models

ACCURACY – ML (PP – 12)

Accuracies of 8 models with raw image matrices, their corresponding histogram vectors

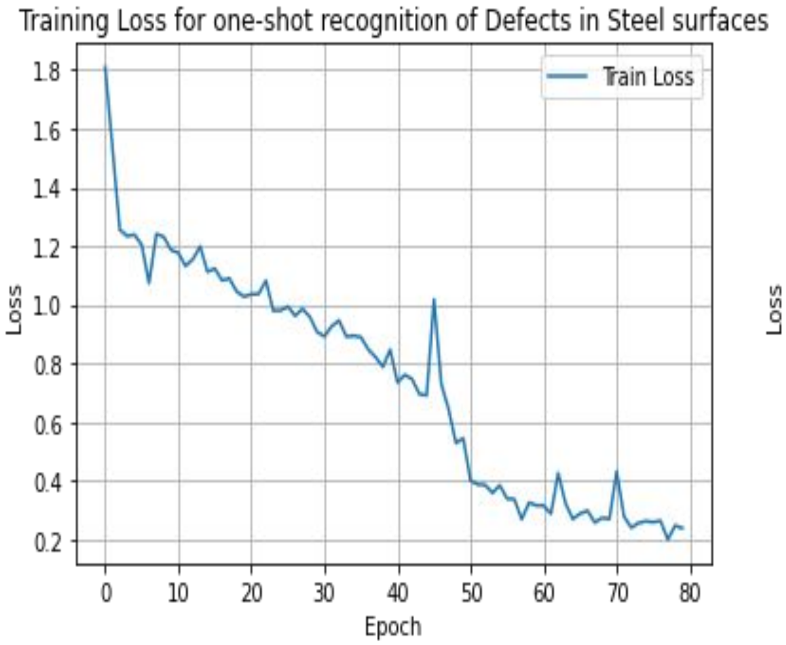


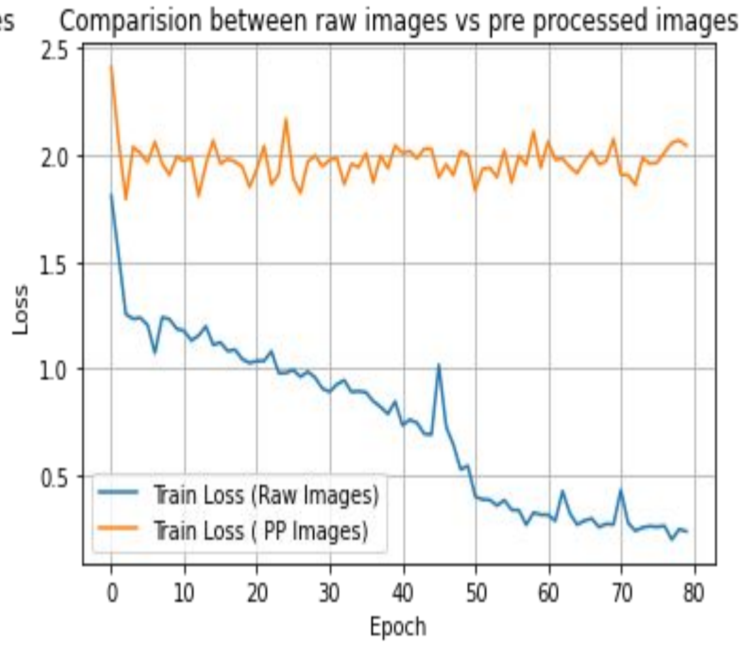
CLASSIFICATION REPORT

```
Confusion matrix for Random forrest classifier
[[69  0  0  6  4  0]
 [ 4 37  3  5  5  4]
 [ 0  4 56  4  3  3]
 [ 6  4  4 63  4  4]
 [ 7  2  0  0 53  4]
 [ 1  5  3  7  1 45]]

              precision    recall  f1-score   support

           1       0.79      0.87      0.83        79
           2       0.71      0.64      0.67        58
           3       0.85      0.80      0.82        70
           4       0.74      0.74      0.74        85
           5       0.76      0.80      0.78        66
           6       0.75      0.73      0.74        62

    accuracy                           0.77       420
   macro avg       0.77      0.76      0.76       420
weighted avg       0.77      0.77      0.77       420

Micro average is :
 0.7690476190476191
```

SVC (Kernel) Classifier with direct pixel values of image as feature vector has performed better than other models

LOSS FUNCTION – SIAMESE CNN

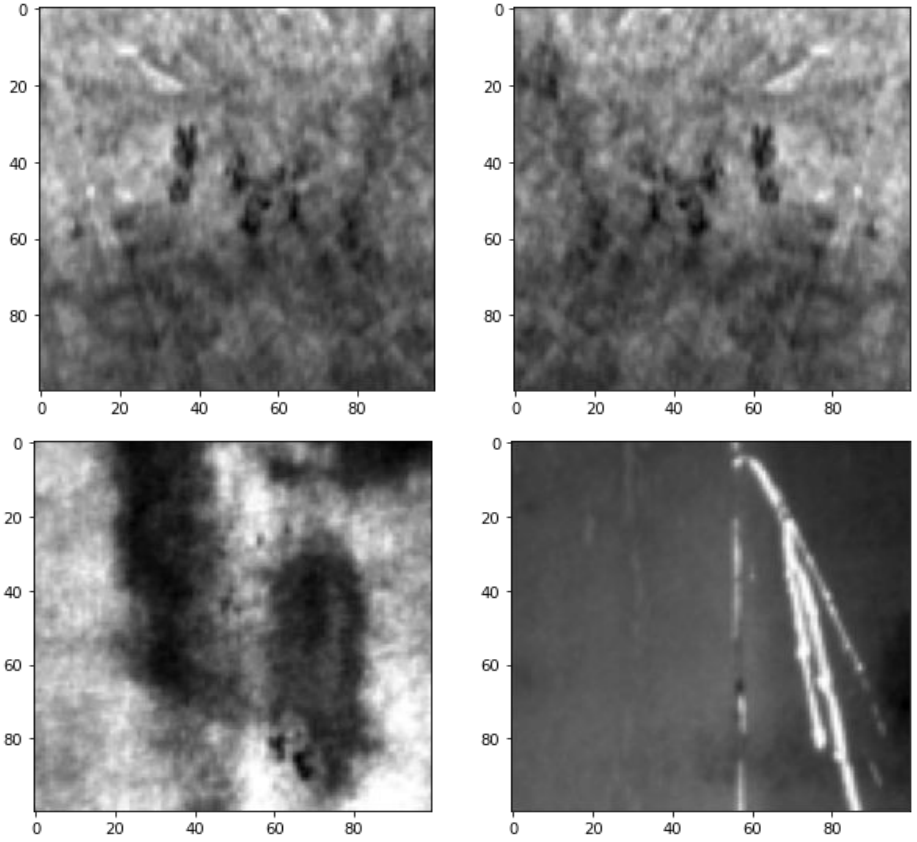




RESULTS – SIAMESE CNN

- 8 random samples have been taken from test set of images to check how the model fared. They have been rotated/ flipped, brightness normalized.
- 7 samples were correctly classified as whether they belong to the same class or not based on the threshold set on the euclidean distance between them, giving an accuracy of 0.875 on raw images
- 6 samples were correctly classified, giving an accuracy of 0.75 on pre processed images

RAW IMAGES



SAME CLASS

Dissimilarity score = 0.044

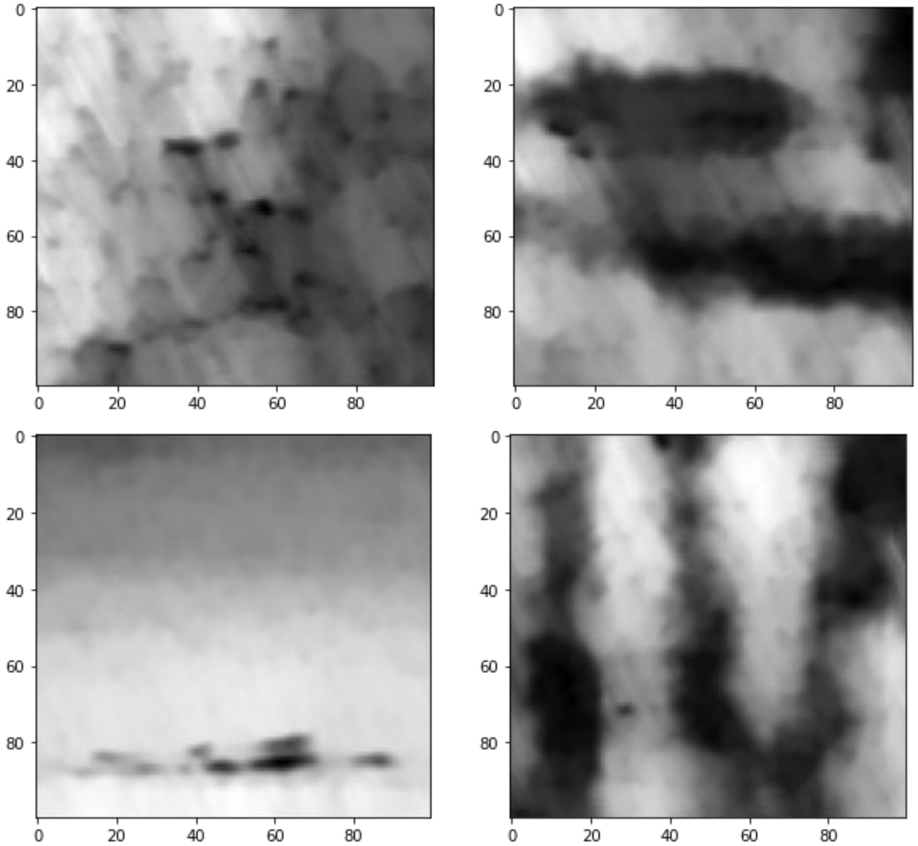
Classified correctly

DIFFERENT CLASS

Dissimilarity score = 2.005

Classified correctly

PRE PROCESSED IMAGES



DIFFERENT CLASS

Dissimilarity score = 1.4

Classified correctly

DIFFERENT CLASS

Dissimilarity score = 0.87

Classified incorrectly

CONCLUSIONS

ASSESSING THE RESULTS

- Overall, Raw images with no preprocessing gave better results than any other experiment with image processing techniques. Preprocessing resulted in loss of features
- Image matrices flattened as one dimensional arrays performed better than histogram feature vectors
- Among traditional ML models, Kernel SVM and random forests showed great potential to deliver better results
- Neural networks detect lines, edges and boundaries through its layers, hence would not require any pre processing

FUTURE SCOPE

- More exhaustive comparison by varying all possible parameters of existing models, and more industry related problems such as time taken to deploy, limited data set
- Turning recognition model into a classification model by extracting feature vectors for each distinct class and comparing with test set images
- Could be extended to modification of control system of machine for real time intelligent machining which can detect the defect in its early stages

REFERENCES

- **One-Shot Recognition of Manufacturing Defects in Steel Surfaces -** Aditya M. Deshpande, Ali A. Minai, Manish Kumar -2020
- **Siamese neural networks for one-shot image recognition - ICML Deep learning workshop** - Gregory Koch, Richard Zemel, Ruslan Zemel, Ruslan Salakhutdinov - 2015 (Core ranking A*)
- https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942
- https://github.com/tensorfreitas/Siamese-Networks-for-One-Shot-Learning
- https://hackernoon.com/one-shot-learning-with-siamese-networks-in-pytorch-8ddaab10340e
- https://medium.com/swlh/one-shot-learning-using-keras-8c1cd62ac382

THANKS!