DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Surface Analysis: Texture Classification of machined metal surface using image processing and machine learning

A Thesis

Submitted by

NAVEEN PRASHANNA G

For the award of the degree

Of

MASTER OF TECHNOLOGY

May 2022

THESIS CERTIFICATE

This is to undertake that the Thesis titled **SURFACE ANALYSIS: TEXTURE CLASSIFICATION OF MACHINED METAL SURFACE USING IMAGE PROCESSING AND MACHINE LEARNING**, submitted by me to the Indian Institute of Technology Madras, for the award of **Master of Technology**, is a bona fide record of the research work done by me under the supervision of **Dr. N. Arunachalam**. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Chennai 600036 **Naveen Prashanna G**

Date: May 2022

Dr. N. Arunachalam
Research advisor
Associate Professor
Department of Mechanical Engineering
IIT Madras

ACKNOWLEDGEMENTS

I am extremely grateful to my thesis guide Dr. N. Arunachalam without whom this research would have never been accomplished. His expertise and guidance helped me through all stages of the projects. I am ecstatic to have had the opportunity to collaborate with him on this project throughout the years.

I would also like to thank the reviewers of the project Dr.P.V.Manivannan and Dr. Sivasrinivasu Devadula for the suggestions which impacted to a positive progress of the project.

I am extremely thankful to Mr. Deep Singh for the continuous support and understanding when undertaking my research and helping my project. Lastly, I would be remiss in not mentioning my parents and friends for their encouragement and support throughout my studies.

Author

Naveen Prashanna G

ABSTRACT

Surface texture image identification from machining surfaces using image processing techniques has become a popular study topic in recent decades. The goal of this work is to use machine learning approaches to recognise diverse machined surface texture photographs. Images of machined components are captured using a camera from mobile device. Two approaches were used. First was to use a Convolutional Neural Network on the surface images to classify the machining process. In second approach, seven statistical characteristics are extracted from collected photos and a feature vector is created. To derive statistical information from machined metal surface pictures, the Grey Level Co-occurrence Matrix is employed. To describe machined surfaces, four machine learning methods were used: Random Forest, Support Vector Machine, Artificial Neural Network, and J48. It is found that the convolutional approach is not learning properly. Artificial Neural Networks and Random Forests have been proven to have 100% training accuracy and 99 percent cross validation accuracy. The collected results illustrate the effectiveness of the GLCM technique, which may be used to recognise texture pictures.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Classification of machined surface is an fascinating discussion in the computer vision and artificial intelligence field and has significance in various fields. Six machining processes Lapping, Reaming, Grinding, Horizontal Milling, Vertical Milling and Turning were chosen for the study. Three sets of parameters for each process and three levels of value of each parameter were taken to make a total of 27 configurations of sets of parameters for each process. Surface images were captured using mobile camera at seven different zoom levels. Mobile camera was used to make the model compatible for daily use without relying on high level tools and setup. Different machine learning approaches were studied to classify the machining process.

Non-contact texture characterisation is regarded as a reliable approach for assessing machined surfaces in industrial sectors, and it is becoming a popular study topic. Computer vision techniques are regarded as a contact free method of acquiring pictures that may be used to determine surface roughness, precisely estimate measurements and detect shapes of complicated components. Various authors presented neural networks, pixel distribution histograms , Gray-level co-occurrence matrix (GLCM), and non traditional mathematical analysis approaches for creating image processing-based surface texture recognition tools.

Several researchers have employed signal analysis, computer vision, and artificial intelligence methods to determine the status of a tool and also the level of wear & tear. Since traditional tool detection systems lack the ability to self-learn, a substantial experimental investigation was undertaken on several machining operations such as turning, milling, and boring to discriminate tool states and anticipate tool wear using artificial intelligence. Machine learning is a popular technique for properly identifying texture images. Many researchers utilise it for classification and regression in a range of applications.

Many writers utilise Naive Bayes (NB) as a classification technique since it is a frequently used machine learning approach. The feature vector is a crucial element for precisely determining the state of any instrument or item. Statistical parameters must be retrieved from the photos in order to create a feature vector. It is thought that if the feature vector is extracted correctly, it will contain all important information. Machine learning methods are then utilised to train, test, and cross validate this feature vector.

Based on the literature that is presently available, it has been discovered that combining computer vision techniques with artificial intelligence is beneficial in the manufacturing industry for the aforementioned purposes: 1) During the examination, no pressure or weight is applied on the sample metal. 2) Sensors or their combination are necessary to analyse the circumstances of the production process, which is costly. 3) When camera pictures are integrated with machine learning algorithms, a quick conclusion regarding the state of the specimen may be made. According to the current literature, the combination of computer vision with artificial intelligence approaches has vast possibilities that have yet to be explored, particularly in industrial applications.

Convolutional neural network model and the impacts of seven texture characteristics retrieved from GLCM for classification of machined surfaces generated from six machining processes are investigated in this work. The experimental results indicate that the suggested GLCM approach will be beneficial for identifying texture features from a variety of machining operations.

CHAPTER 2

MACHINING PROCESSES

Cutting parameters, tool geometry, work-piece material, chatter, and cutting fluids all contribute to surface roughness. Of these factors, three important parameters are chosen for each machining process. Three levels of values for each parameter were taken. By varying the combination of parameters and levels, 27 configurations of sets of parameters are obtained for each process. In this experiment, six processes were studied which we will see below. Hardened Steel was used for the machining processes.

2.1 LAPPING

Lapping is a machining operation that involves rubbing two surfaces together with an abrasive between them, by using machine. Copper plate was used for the lapping process. A three-factor, a three-level orthogonal experiment was designed and the specific experimental parameters and levels are shown in Table 2.1.

Parameters	Levels		
	1	2	3
Lapping Normal Pressure (kg)	5.0	10.0	15.0
Abrasive Size (μm)	1.0	3.0	5.0
Abrasive Concentration (wt.%)	0.5	1.0	2.0

Table 2.1: Machining Parameters and their levels for Lapping process

2.2 REAMING

Reaming is a cutting technique that includes using a rotary cutting tool to smooth the inside walls of a workpiece's existing hole. Three process parameters, each at three levels, are chosen and shown in Table 2.2.

Parameters	Levels		
	1	2	3
Speed (rpm)	900	1200	1500
Feed Rate (mm/min)	90	135	180
Reverse Feed Rate (mm/min)	5000	7500	10000

Table 2.2: Machining Parameters and their levels for Reaming process

2.3 GRINDING

Grinding is the method of removing metal with abrasives melded together to form a spinning wheel. When the grinding wheel make contact with the workpiece, the abrasive particles operate as miniature cutting tools, each one removing a small chip from the workpiece. Machining parameters chosen for this process are Grinding wheel speed, workpiece speed and depth of cut. Their values are shown in Table 2.3.

Parameters	Levels		
	1	2	3
Grinding Wheel Speed (m/min)	1600	1800	2000
Workpiece Speed (rpm)	150	200	250
Depth of cut (mm)	0.02	0.04	0.06

Table 2.3: Machining Parameters and their levels for Grinding process

2.4 HORIZONTAL MILLING

Machines for horizontal milling install the cutting tool on a horizontally oriented spindle that may remove material from the stationary workpiece selectively. Three process parameters chosen for this process and their levels are shown in Table 2.4.

Parameters	Levels		
	1	2	3
Spindle Speed (rpm)	750	1000	1150
Feed Rate (mm/min)	50	100	175
Depth of Cut (mm)	0.4	1.0	1.5

Table 2.4: Machining Parameters and their levels for Horizontal Milling process

2.5 VERTICAL MILLING

A vertical milling machine has a vertically oriented spindle that keeps the cutting tool against the stationary workpiece while also rotating it. As the workpiece is stationary, the spindle moves towards it and pushes against it. There are 27 runs for the design and all the parameters are shown in Table 2.5 below.

Parameters	Levels		
	1	2	3
Spindle Speed	2000	3000	4000
Feed Rate (mm/rev)	0.02	0.04	0.06
Axial Depth of Cut (mm)	1.5	2.5	3.5

Table 2.5: Machining Parameters and their levels for Vertical Milling process

2.6 TURNING

During the turning process, a cutting tool removes material from the external diameter of a spinning object. In this operation, a cutting tool, usually a non-rotary tool bit, moves linearly while the workpiece spins to create a helical tool path. The machining parameters and their levels are shown under Table 2.6.

Parameters	Levels		
	1	2	3
Spindle Speed	780	1560	2340
Feed Rate (mm/rev)	1.21	1.81	3.63
Depth of Cut (mm)	0.5	1.0	1.5

Table 2.6: Machining Parameters and their levels for Turning process

CHAPTER 3

DATA COLLECTION AND DATA AUGMENTATION

3.1 DATA COLLECTION

Input images for the model are created by manually taking pictures of the machined surface using a mobile phone. Pictures are taken at 7 different zoom levels for each set of parameters for each process. Camera and the metal work-piece are surrounded by white cardboard to avoid reflection of surroundings. Even the camera angle is adjusted so that even the reflection of the camera on the metal surface is avoided. Then the pictures are manually cropped so that it contains only the machined metal surface.

The resulting picture is transformed to a Gray scale level (ranging from 0 to 255), which will aid in further investigation. By using Python OpenCV software program, the obtained original RGB image is converted into gray scale image. These images are further cropped into square images of size 224x224.

3.2 DATA AUGMENTATION

Data augmentation is a term used in data analysis to describe methods for extending the size of a dataset by appending slightly modified copies of current datapoints or by generating new artificial data from old data. It acts as a preprocessing step and helps to prevent overfitting while training a classification model. It is strongly connected to data analysis oversampling.

In our experiment, three techniques[Horizontal flip, Vertical flip and Image Rotation] were used. In horizontal flip, images are flipped left to right, while in vertical flip, images are flipped top to bottom. Then copies of those images are rotated at 90° to the axis passing through the image to create further samples of images. To perform a image rotation, we need to create a rotation matrix and then apply affine transform to it. About 100 thousand images were created combining both the processes after the augmentation process.

```
import cv2
def horizontal_flip(img):
    return cv2.flip(img, 1)

def vertical_flip(img):
    return cv2.flip(img, 0)

def rotation(img, angle):
    h, w = img.shape[:2]
    M = cv2.getRotationMatrix2D((int(w/2), int(h/2)), angle, 1)
    img = cv2.warpAffine(img, M, (w, h))
    return img
```

CHAPTER 4

APPROACH USING DEEP LEARNING CNN MODEL

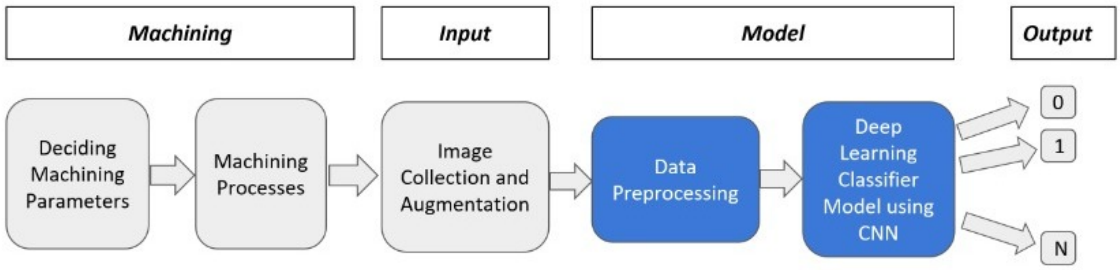
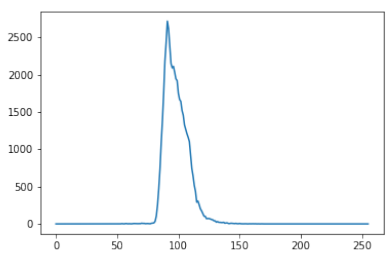


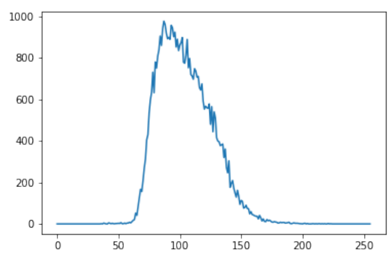
Figure 4.1: Approach using Deep Learning CNN model

4.1 DATA PRE-PROCESSING

The images of size 224x224 pixels are feed into the neural network. The image histogram of grayscale square images is obtained to understand the level of pixels in the images. Figure 4.2a indicates that the pixels distribution is not equal at all the points. So, the histogram sliding is required to enhance the captured image for further studies. As seen in Figure 4.2b, a constant value is applied to all pixels in the picture, shifting the histogram to the right. This increases the brightness of the image.



(a) Before histogram sliding



(b) After histogram sliding

Figure 4.2: Histogram of metal surface image

The low frequency pixels in the image acquired after histogram sliding are eliminated using a high pass filter. The high pass filter will not modify the signal's high frequency pixels, but it will dampen the signal's low frequency pixels and remove any bias. The scratches on the surface are plainly visible for additional processing since the edges are intensified using a high pass filter. Finally, the image is passed through a binary threshold filter which converts every pixel to either black or white with some threshold value. Images of machined surfaces during all stages of preprocessing methods are shown in the Figure 4.3

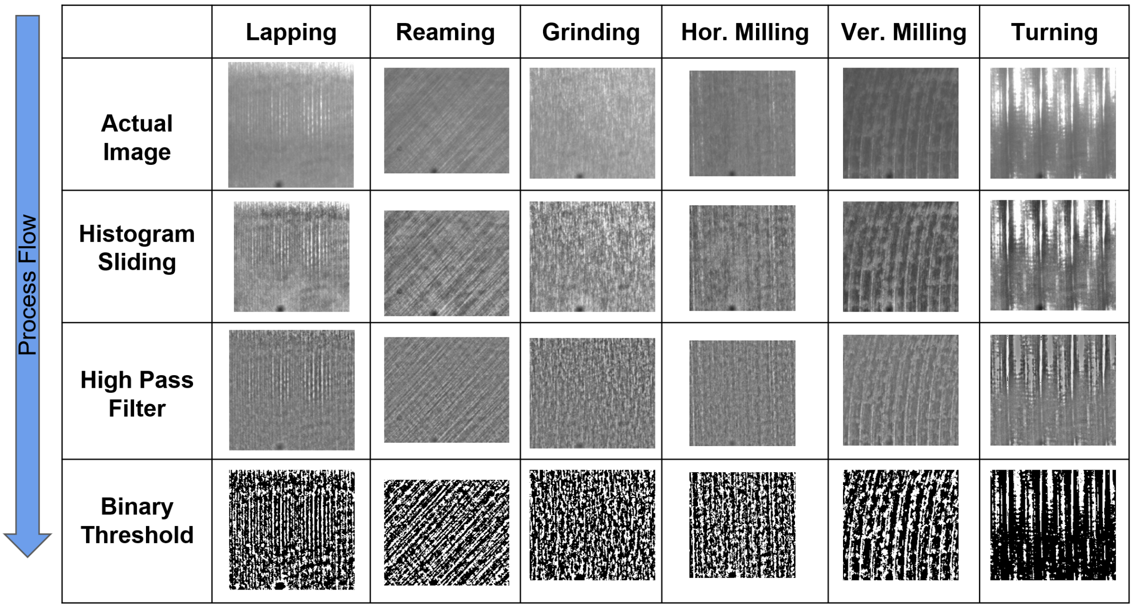


Figure 4.3: Preprocessing images of all machining processes

4.2 CONVOLUTIONAL NEURAL NETWORK MODEL

The CNN model is a form of neural network that allows us to extract higher representations for picture input. Unlike traditional image recognition, which requires the user to define the image characteristics, CNN takes the picture's raw pixel data, trains the model, and then extracts the features for improved categorization. The window is swept across pictures using a convolution, which then calculates the input and filter dot product pixel values. This enables convolution to highlight the important characteristics. This calculation detects a certain feature in the input picture and generates feature maps (convolved features) that highlight the most significant characteristics.

The dataset is divided into three parts: an 80% training set, a 10% validation set, and a 10% test set. ImageDataGenerators are used to read the images as we have a large amount of images in the dataset. A CNN model which has a input layer, 3 convolution blocks, a ANN block with an output layer is built to classify the images. The model is compiled using a categorical_crossentropy as loss function & adam optimizer and evaluated using accuracy as the metrics.

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense
from keras.layers import InputLayer, BatchNormalization, Dropout
# build a sequential model
model = Sequential()
model.add(InputLayer(input_shape=(224, 224, 3)))
# 1st conv block
model.add(Conv2D(25, (5, 5), activation='relu', strides=(1, 1)))
model.add(MaxPool2D(pool_size=(2, 2), padding='same'))
# 2nd conv block
model.add(Conv2D(50, (5, 5), activation='relu', strides=(2, 2)))
model.add(MaxPool2D(pool_size=(2, 2), padding='same'))
model.add(BatchNormalization())
# 3rd conv block
model.add(Conv2D(70, (3, 3), activation='relu', strides=(2, 2)))
model.add(MaxPool2D(pool_size=(2, 2), padding='valid'))
model.add(BatchNormalization())
# ANN block
model.add(Flatten())
model.add(Dense(units=100, activation='relu'))
model.add(Dense(units=100, activation='relu'))
model.add(Dropout(0.25))
```

```
# output layer
model.add(Dense(units=6, activation='softmax'))

# compile model
model.compile('categorical_crossentropy', "adam", ['accuracy'])
# fit on data for 5 epochs
model.fit(train , epochs=5, validation_data=val)
```

4.3 RESULTS

To fit the training set, batch size 32 is used to feed it into the model. Validation set was also fed in batches of size of 32. It gave a accuracy of 99.1% and 45.37% on the training and validation set respectively in 5 epochs. This model is used to predict the test set images on which it gave a accuracy of 45.5%. The confusion matrix for the prediction on test set is shown in the Table 4.1.

		Predicted					
		Lapping	Reaming	Grinding	H.Milling	V.Milling	Turning
True	Lapping	100	0	0	26	180	4
	Reaming	0	0	0	320	0	0
	Grinding	0	0	0	0	552	0
	H.Milling	0	0	0	408	204	0
	V.Milling	11	0	4	12	526	64
	Turning	57	0	2	118	81	331

Table 4.1: Confusion matrix for CNN model

CHAPTER 5

APPROACH USING GRAY LEVEL CO-OCCURENCE MATRIX

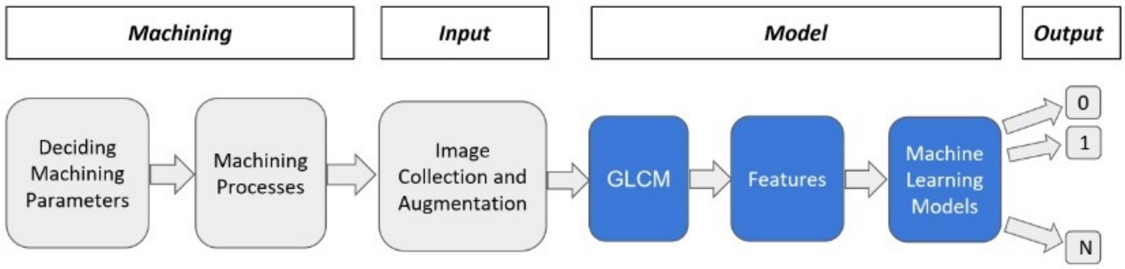


Figure 5.1: Approach using GLCM and ML model

5.1 GRAY LEVEL CO-OCCURENCE MATRIX

The relationship between the pictures and pairs of pixels is discovered using GLCM, a statistical approach. GLCM transforms an image into a matrix with the same number of rows and columns as the total number of pixel values in a surface picture which in this case is 256. The GLCM matrix represents the occurrence of one grey level appearing in a preset geometric connection with another grey level within the study region. Twelve variations of GLCM were created by varying distances and angles between the pixel pair.

Figure 5.2 depicts the process of using a grey co-matrix to compute a few values in the GLCM of a 4-by-5 picture. Component (1, 1) has the value 1 in the GLCM because there is just one and only situation in the picture where two pixels on a level plane adjoining each other have the characteristics 1 and 1. Because there exists two occurrences in the matrix which has 2 evenly neighbouring pixels exhibiting the characteristics 1 and 2, cell (1, 2) in the GLCM has the value 2. Since there exists no occurrences of nearby pixels with the characteristics 1 and 3, cell (1, 3) in the GLCM has the value 0.

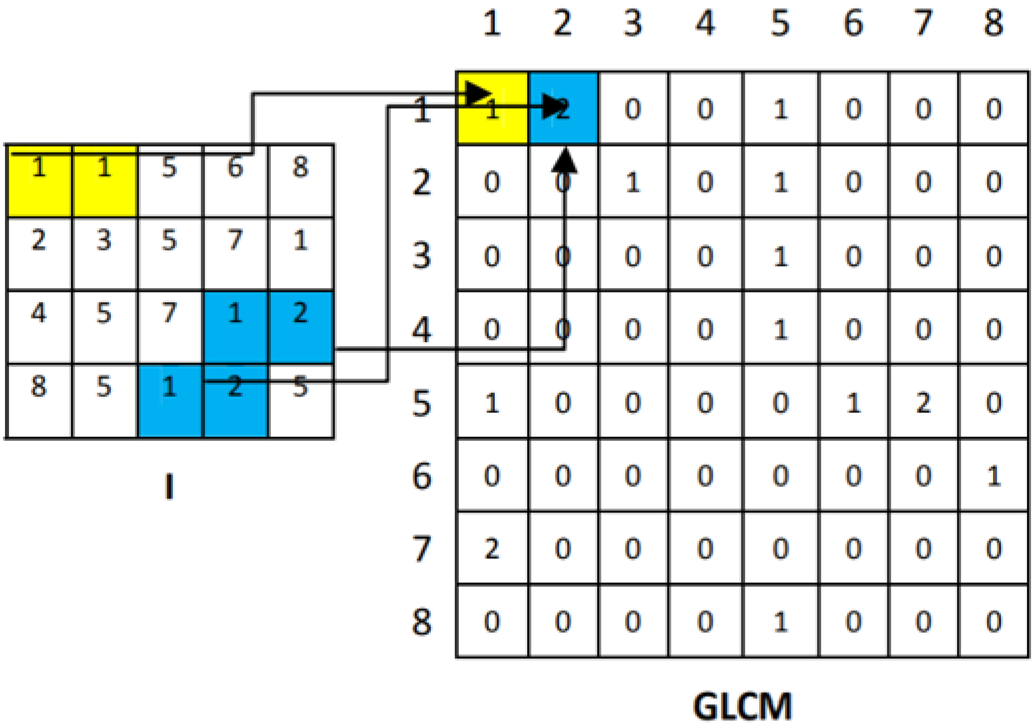


Figure 5.2: Transformation of image into GLCM

5.2 FEATURE EXTRACTION FROM GLCM

Surface feature extraction is the process of extracting features from pictures of metal surfaces using the characteristics and look of the object which is again determined by the roughness, cutting lines, etc. The GLCM is used to compute seven surface characteristics in the current project, which are listed below. With the 12 variations in GLCM and 7 features in each GLCM, we get a feature vector of size 84.

$\rightarrow Contrast : \sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$ $\rightarrow Dissimilarity : \sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$

$\rightarrow Homogeneity : \sum_{i,j=0}^{levels-1} P_{i,j}/(1+(i-j)^2)$ $\rightarrow Entropy : -\sum_{i,j=0}^{levels-1} P_{i,j} log(P_{i,j})$

$\rightarrow SecondMoment(ASM) : \sum_{i,j=0}^{levels-1} P_{i,j}^2$ $\rightarrow Energy : \sqrt{ASM}$

$\rightarrow Correlation : \sum_{i,j=0}^{levels-1} P_{i,j}\{((i-\mu_i)(j-\mu_j))/\sqrt{(\sigma_i^2)(\sigma_j^2)}\}$

5.3 MODELS

In surface classification, the pictures are categorised according to the machining processes when a collection of extracted data from pictures, referred as a feature vector, is passed into classification models. While training, the algorithm converts the input data extracted from the pictures to the output label, and it classifies the pictures while testing. In this project, Support Vector Machine(SVM), Decision Tree J48, Random Forest and Artificial Neural network (ANN) are used to classify the texture images. Feature vector obtained from GLCM is fed into each model to train the model and accuracy for each model is obtained.

5.3.1 Support Vector Machine

Support Vector Machine is widely used to solve classification problems. It finds an hyperplane that separates the dataset into their classes. SVM employs support vectors to improve the accuracy of a given data set by categorising it into distinct categories. This support vector is located on the ideal hyper plane by the greatest margin. The optimum hyperplane divides datapoints from one class from datapoints from another, yielding a classification result.

SVM models function as follows for multi-class classification problems: Let N indicate the entire number of classes, and N1 is trained as a new class while the remainder of the class is trained with another class. N1 is deleted in the next round, and N2 is regarded a new class, with the remaining classes being considered second classes. For multiclass data sets, the process will be repeated for remaining classes, with the mean result serving as the final result.

```python
from sklearn.svm import SVC
clf = SVC()
clf.fit(X_train, y_train.ravel())
y_pred = clf.predict(X_test)
```

```python
print(classification_report(y_test, y_pred,digits=5))
print(confusion_matrix(y_test, y_pred))
print("Accuracy_of_the_Model:", accuracy_score(y_test, y_pred))
```

		Predicted					
		Lapping	Reaming	Grinding	H.Milling	V.Milling	Turning
True	Lapping	927	0	9	0	1	63
	Reaming	0	1000	0	0	0	0
	Grinding	13	0	921	66	0	0
	H.Milling	0	0	5	994	1	0
	V.Milling	4	0	7	18	968	3
	Turning	31	0	3	1	0	965

Table 5.1: Confusion matrix for SVM model

5.3.2 Decision Tree J48

The J48 method is a supervised technique for data classification. J48 is a variant of the C-4.5 and ID-3 algorithms, which have been commonly utilised for decision tree building. For the examination of datasets given by the user, decision trees used a graph-based branching technique. The internal node represents an attribute test, the branch represents test results, and the leaf node represents class labels. Internal nodes divide samples into subsets based on distinct functions of attribute values in order to determine a class. Samples are categorised by leading them from the root to a leaf in the tree, with each leaf being given to one category that reflects the most suitable outcome. The process is continued until the final outcome of the category classification decision is attained.

```python
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train.ravel())
y_pred = clf.predict(X_test)
```

```python
print(classification_report(y_test, y_pred, digits=5))
print(confusion_matrix(y_test, y_pred))
print("Accuracy_of_the_Model:", accuracy_score(y_test, y_pred))
```

		Predicted					
		Lapping	Reaming	Grinding	H.Milling	V.Milling	Turning
True	Lapping	85	0	4	0	2	9
	Reaming	0	998	0	1	1	0
	Grinding	7	0	968	23	1	1
	H.Milling	0	0	17	975	8	0
	V.Milling	7	2	11	10	964	6
	Turning	15	0	1	2	3	979

Table 5.2: Confusion matrix for Decision Tree model

5.3.3 Random Forest

Breiman L. invented the Random Forest classifying method. It is an ensemble approach that use a tree as a predictor. To enhance classification, Random Forest builds a vast amount of decision trees from a dataset based on bagging. Bagging is a technique for reducing dataset variation while avoiding overfitting. In the beginning, the training set was split into in-set and out-set bags. The decision tree is generated for every in-set bag of dataset, and the classification accuracy is evaluated using the out-set bag of dataset. The trees created using the out-set bag from the complete training data set yield final results.

```python
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators = 100)
clf.fit(X_train, y_train.ravel())
y_pred = clf.predict(X_test)
```

```
print(classification_report(y_test, y_pred, digits=5))
print(confusion_matrix(y_test, y_pred))
print("Accuracy_of_the_Model:", accuracy_score(y_test, y_pred))
```

		Predicted					
		Lapping	Reaming	Grinding	H.Milling	V.Milling	Turning
True	Lapping	994	0	0	0	0	6
	Reaming	0	1000	0	0	0	0
	Grinding	4	0	985	11	0	0
	H.Milling	0	0	2	996	2	0
	V.Milling	0	0	1	7	991	1
	Turning	1	0	0	0	0	999

Table 5.3: Confusion matrix for Random Forest model

5.3.4 Artificial Neural Network

After transforming input feature vector into training and testing data sets, Artificial Neural Network (ANN) is a technique for classification using feature vectors. ANN is used to tackle issues in pattern recognition, manufacturing, forecasting, robotics, and other fields. Various layers of linked nodes make up a neural network. The feature vector is sent into the algorithm via an input layer that contains at least one hidden layer where the actual data preparation is done.

The input feature-vector is made up of n degree feature-vectors, with n being the overall number of characteristics (features). First hidden layer has 256 nodes and second layer has 64 nodes. All the nodes are densely connected and has 'relu' as the activation function. The output layer has six nodes for six corresponding processes, so that every information of a specific class is retained. The categorization results are reflected in the set of nodes linked to the output layer. The weightage acquired by each instance based on provided qualities is used to make a classification decision.

```python
import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint

ann = tf.keras.models.Sequential()
# Adding layers
ann.add(tf.keras.layers.Dense(units=256,activation="relu"))
ann.add(tf.keras.layers.Dense(units=64,activation="relu"))
ann.add(tf.keras.layers.Dense(units=6,activation="sigmoid"))

ann.compile("adam","binary_crossentropy", ["accuracy"])
ann.fit(X_train, y_train, batch_size=32,epochs=50)

y_pred = ann.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)+1
print(classification_report(y_test, y_pred, digits=5))
print(confusion_matrix(y_test, y_pred))
print("Accuracy_of_the_Model:", accuracy_score(y_test, y_pred))
```

		Predicted					
		Lapping	Reaming	Grinding	H.Milling	V.Milling	Turning
True	Lapping	1000	0	0	0	0	0
	Reaming	0	1000	0	0	0	0
	Grinding	0	0	994	6	0	0
	H.Milling	0	0	17	983	0	0
	V.Milling	0	0	1	0	999	0
	Turning	0	0	0	0	0	1000

Table 5.4: Confusion matrix for ANN model

5.4 RESULTS

Support Vector machine gave a accuracy of 96.25% on the test test and Decision Tree model gave a accuracy of 97.81%, while both Random forest and ANN models gave a accuracy of more than 99% on the test set. The ANN model is performing well with a accuracy of 99.6%. The confusion matrix for SVM, Decision Tree, Random forest and ANN is shown in Tables 5.1, 5.2, 5.3 and 5.4 respectively.

CHAPTER 6

CONCLUSION

- The use of computer vision techniques and artificial intelligence algorithms to identify surface texture pictures of machined surfaces is investigated in this work.

- Computer Vision techniques emerge as a potential decision-making tool for automated inspection.

- Texture analysis has the advantage of being a non-contact inspection method that can be used for observing tool condition, roughness evaluation, defect sensing, and so on.

- Convolutional Neural Network approach was able to get a 99% accuracy on the training set. But it only gave a accuracy of 45% on the validation set. This shows that the network is overfitting on the training set. It couldn't learn the surface features properly.

- On the other hand, all the models in the GLCM approach gave accuracy more than 95%. This shows that the features extracted from the GLCM matrix is highly co-relates to the actual surface features of the image.

- The methods utilised for feature extraction from GLCM and artificial intelligence algorithms imply a possible use case in industries for the progress of real time characterization of pictures from diverse industrial operations and services, according to the experimental results.

REFERENCES

[1] Jiayun Deng et al. "Optimisation of Lapping Process Parameters for Single-Crystal 4H–SiC Using Orthogonal Experiments and Grey Relational Analysis". In: *Micromachines* 12 (July 2021), p. 910.

[2] Harne Shingarwade and MS Harne. "Effect of Reaming Process Parameters on Surface Roughness Using Taguchi Method". In: *International Journal of Engineering Research & Technology* 1 (2013), pp. 362–365.

[3] Sandeep Kumar and Onkar Bhatia. "Experimental Analysis Optimization of Cylindirical Grinding Process Parameters on Surface Roughness of En15AM Steel". In: *Journal of Mechanical Engineering* Volume 12, (July 2015).

[4] Mrs Londhe/Chilwant. "Optimization of cutting parameters in Milling Operation to improve surface finish of EN 31". In: (Jan. 2016).

[5] V V K Lakshmi. "Modelling and Optimization of Process Parameters during End Milling of Hardened Steel". In: *International Journal of Engineering Research* volume2 (Mar. 2012), pp. 674–679.

[6] Rahul Davis, Vikrant Singh, and Shaluza Priyanka. "Optimization of Process Parameters of Turning Operation of EN 24 Steel using Taguchi Design of Experiment Method". In: *Lecture Notes in Engineering and Computer Science* 2 (July 2014), pp. 1045–1047.

[7] Srivani A and Anthony Xavior M. "Investigation of Surface Texture Using Image Processing Techniques". In: *Procedia Engineering* 97 (2014), pp. 1943–1947. ISSN: 1877-7058.

[8] Julian Balcerek et al. "Classification of road surfaces using convolutional neural network". In: (Sept. 2020), pp. 98–103.

[9] Dhiren Patel, Vinay Vakharia, and M.B. Kiran. “Texture classification of machined surfaces using image processing and machine learning techniques”. In: *FME Transactions* 47 (2019).

[10] D. Nathan, G. Thanigaiyarasu, and Kogi Vani. “Study On the Relationship between Surface Roughness of AA6061 Alloy End Milling and Image Texture Features of Milled Surface”. In: *Procedia Engineering* 97 (Dec. 2014).

[11] S I Mohamed Suhail et al. “Vision based system for surface roughness characterisation of milled surfaces using speckle line images”. In: *IOP Conference Series: Materials Science and Engineering* 402 (2018), p. 012054.