Neural Network Based Image Classifier for Nuts and Bolts

Faizan Muneer, Mtech Student Department of Mechanical Engineering, RIMT University Mandi Gobindgarh, Punjab India,faizanmuneer24@gmail.com

Amritpal Singh , Assistant Professer, Department of Mechanical Engineering, RIMT University, Mandi Gobindgarh, Punjab India, amritpal.singh@rimt.ac.in

Abstract The fundamental objective of the work is to develop a system that will be beneficial in the mechanical sector for the recognition of bolts and nuts. The aim of the study is to develop an image processing algorithm using principal component analysis to deliver standardized pictures which can be utilized as inputs for processing and detection. The Spyder software integrates all algorithms. This application also warrants a prototype which replicates the triage of nut and bolt. The input is a properly pre-processed photo. The picture is exposed to major component analysis for feature extraction (PCA).The result is inserted into a neural artificial network (ANN). This allows things to be reliably recognized and answered when required.

Keywords —Artificial intelligence, Bolt, Convolution Neural Network, Deep Learning, , Nut, Spyder Software

I. INTRODUCTION

Humans can recognize any object by the natural logical thinking process by human perception of identification. But machines are not having the perception of identification for recognizing of an object as it is far away from the human recognition system. So research is to be developed for increasing the recognition efficiency of the machine A challenging task is to authenticate determination of the nuts and bolts during its production. So the determination parameters are design depending on the size quality of the nuts and bolts. But these parameters are changes or vary from machine to machine. Machines are made up of various parts to extract the image from an object to solve the same task. To solve the problem, a branch of engineering and technology is used called as computer vision. Computer vision is a technique use to extract the information from the various images based on automated approach useful in the automobile industries. In computer vision, a high performance camera performs the function of human eye which minimizes the human biological system in visual system. Camera is used to capture the targeted images for this system.[1] These images are then stored in the memory of computer which are further used for process of recognition. We shall be using spyder software throughout this project, For the sorting of nuts and bolts on a conveyer belt in mechanical industries, the wavelet transform technique is used.

Artificial neural network is technique which works like a human brain to retain the data as it is first identified or trained. The main objective of this project to study the various feature extraction technique and then finalized the one of the technique for this project for getting faster and perfect result.

In this paper, we use the spyder software with keras and tenserflow 2.0 and implement the principle component analysis and artificial neural network for image processing and detection Keras is a high-level API for TensorFlow 2.0: a user-friendly, highly productive interface to deal with machine learning problems with the present concentration on deep learning. It offers the required abstractions and building pieces for machine learning systems development and deployment.[8][9]

II. LITERATURE REVIEW

In view-based techniques, the object of interest (OOI) and the precalculated 2D prospects of the object are compared. This approach can provide the object's position an approximate estimation, but owing to the wide geographical area searched it is computation-intensive. Cyr and Kimia[7] have presented a way to tackle this challenge, which collected related OOIs in one class, and generated similarities for identification. Eggert et al. [2], Ulrich et al. [4] and Steger et al. [3] enhanced the aspect graphs and similarity metric. However, this strategy was often not used because of its great degree of complexity

. BORDER was invented by Jacob, an abbreviation for the Enclosed Region Bounded Oriented Rectangle Descriptors, where they enclose the object in a rectangle to find minimal outliers. It outperformed photos in contrast to previous state-of-the-art BOLD and Line2D Descriptors for

 DOI : 10.35291/2454-9150.2022.0400

occlusive pictures and confusion. It divides the portion of the line into tiny, equal parts. These lines are then placed in rectangles to cover the inner area of the item. The rectangle is rotated at different angles and produces multiple rectangular layers to further lessen the occlusion effect. The three-stage framework of the SIFT approach is followed by detection, description and correspondence.

The count of textureless objects on a scene remained a problem until a technique was presented by Verma to count things using the form and colour attributes. First of all, morphological border removal and segmentation using mean hue value (colour properties) and Hu-moments during preprocessing were used (shape features). The objects were then classified by means of SVM, kNN, NN and tree bagging. They found that the tree lining had the highest precision in experiments for all classifiers. The categorised objects will finally be shown.

Finally, the categorised items were tallied with bounding boxes around them. In 2016, Thoduka and others created a technology which employed RGB-D-based characteristics to differentiate items such as nuts, pins, big, tiny black profiles and so on. They found the 3D point cloud of all objects in the RGB-D collection using live 3D camera photographs. The bounding box, the medium/medium colour, the cloud dot size and the circularity have been constructed and sent into an SSM classifier. They concluded that the addition of more characteristics did not increase precision.

In order to solve previous technical weaknesses, BIND [5] was established in 2017. It is a detector which employs several binary network layers to reliably detect textureless objects. First the object is embedded in a network that is then divided into identical fixed chunks.

These blocks offer information on the body of the object in the form of binary bits. The main benefit of employing BIND is that it invariably applies to characteristics such as scaling, rotation and translation. In order to fit the template to the real picture, AND & OR operators are bitwise employed.

Xiang has built a state-of-the-art Pose CNN, a converting network for object prediction in different orientations. The object distance is calculated from the camera in order to predict the item's translation matrix. The rotation matrix of the object is developed.

Three major functionalities exist for Pose CNN.

I It allocates to each item in the collection, first and foremost, a semantic label.

ii) The distance between the centre of the item and the centre of the picture is then calculated.

iii) 3D postures may be approximated because of the inherent knowledge of the camera.

Finally, a representation of quaternion is employed to predict 3D rotation.

In mid-2019 Park et al. [6] proposed a system that uses a profound picture as input and creates an area of interest (ROI). ROI is a patch in which the network thinks there is an object or an object. The ROI is then reduced to 14x14 with 256 filters and retrieved further descriptors. This ROI is then compared to the ROI from the picture segmentation and posture estimation template pre-calculated.

III. METHODOLOGY

The whole procedure of developing a Nut Recognition Model using deep CNN is more extensive. The whole technique is broken down in the following paragraphs, starting with a photo collection for the phase of classification of the deep neural network.[11]

- Dataset

  Appropriate datasets are required at every level of object recognition research, from the education phase to performance assessments of reconnaissance algorithms. All the photographs in the collection are found on the web and are checked for the use of names for diseases and plants. An additional class is introduced to the dataset to find healthy leaves. It will be used only pictures of healthy leaves. An extra class in the dataset with backdrop pictures is beneficial for more accurate categorization. A deep network of neurons may be taught to differentiate between the leaves and the backdrop.

- Image Pre-processing and Labelling

  Images may be obtained through the Internet in a number of formats, resolutions, and degrees of quality. Final photographs for use as a data set for a deep neural network classifier will be pre-processed to increase consistency and extraction of features. In addition, picture preprocessing processes such as manual cuts to highlight the region of interest all photographs (plant leaves).

- Neural Network Training

To develop an image classification model, use a data packet to train a deep convolutionary neural network. Artificial NNs are two of the most prominent technology for the detection of plant diseases now in use. Supporting Vector Machines include (SVMs). They are used with various types of picture preprocessing to increase the extraction of features.

 DOI : 10.35291/2454-9150.2022.0400

- Test

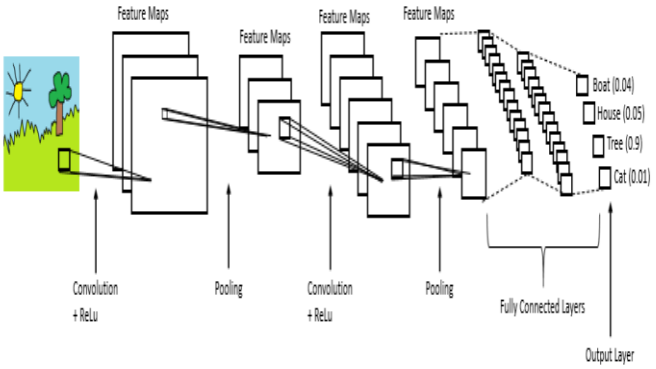
Splitting data into a workout and a test set and then training a neural network on a training set and using the predicted test set is a usual approach of evaluating artificial neural network performance. The accuracy of our predictions can be calculated since both the initial and predicted results for the test set are known[12].

CNN architecture was built to reflect the connectivity network of neurons in the human brain, based on the organisation and the functions of the visual cortex.[10]

In a CNN, each neuron group is divided into a three-dimensional frame, each of which examines a small section or feature of the image. In other words, each pair of neurons aims at recognising a certain component of the picture. The ultimate CNN output is a vector with probability values which shows the probability that a certain charact belongs to a certain class.

Working of CNN-The CNN layers

- A multi-layered CNN is made of:
- Convolutionary layer: Applies a scan filter to build a feature map that predicts the class probabilities of each feature, a couple of pixels at a time. A convolution layer is the essential component of the CNN architecture, which generally includes a combination of linear and non-linear processes.
- Downsample pooling layer: Reduces the amount of data generation for each function by the overlay layer and maintains the most significant data. A batching layer conducts a standard sample operation on function maps, reduces its in-plan aspect ratio, leads to small shifts and distortions in translation invariance and reduces number of learning parameters. Note that none of the pooling layers has learnable parameters; instead, filter size, step and pads, similar to convolution processes, are hyperparameters in pooling. Fully linked input layer: converts the outputs from previous layers into a single vector that may be utilised as an input for the next layer.
- Fully linked input layer: turns the previous level outputs into a single vector that may be used as the next layer input
- Fully connected layer: provides a precise label with weights applied to the input collected from the analysis functionality.
- Completely connected output layer: compute the ultimate probability of picture class determination



. Fig 1 **Basic CNN Architecture**

IV RESULTS AND FINDINGS

Implementing code

Import dataset and load the libraries

Firstly we import all the modules that are needed to train our model. Following are the imported modules used in training:

- Image Data Generator:

In batches that improve data in real time, Image Data Generator generates tensor image data. The output images of the generator are given the same output dimensions as the input photographs.
The image data increase is a technique for artificially extending the dimensions of a training dataset by generating modified picture versions.
Deep learning neural network models trained on more data are better qualified, and techniques to enhancement can provide differences in pictures that boost the ability of fit models to generalise the lessons learnt from new pictures.

- Conv2D:

By converting the layer input with a convolution kernel, this layer creates a tensor of outputs. A matrix or mask can be employed to blur, sharpen, emboss, identify the edge of an image and to execute a convolution between the kernel and the image.

- MaxPooling2D:

The highest concentration is used for 2D spatial data. Proof the input display by using the maximum value on the feature axis for each dimension across the pool size frame. The window is modified by steps in each dimension.

- Dropout:

The dropout layer that minimises overfit is used to randomly change the input units to 0 at a predetermined frequency during every stage of the training. Electronically specified inputs not set to 0 are 1/(1 -rate) up to keep the overall sum the same. No values will be dropped throughout the lesson if training is set to true and the

 DOI : 10.35291/2454-9150.2022.0400

dropping layer is applied. Training will automatically be set to True when using model.fit, and in other circumstances you may set the kwarg explicitly to True when calling the layer. In our training model, we employed a dropout of 0.25.

- Dense:

The dense layer is the regular, interconnected neural network layer. This layer is the most often used and popular.

- Flatten:

The entrance is flattened. The lots are unchanged. Between the convolutionary and completely connected layers is a "Flatten" layer. A dual matrix of features is placed in a vector which may be entered into a completely connected neural network classifier.

- EarlyStopping:

The number of training periods to be employed is challenging to train neural networks. To exceed the training data set can arise from too many epochs, while using too few might lead to a flawed model. Early stoppage is a strategy which allows you indicate an arbitrary amount of training periods, and then end training if the performance of the model stops increasing on the overall validation data set. In our railway simulation, we employed patience=10

- Sequential:

It is straightforward to develop deep learning models with the Keras Python Module. For most problems, you can layer by layer create models with the sequential API. You can't construct models with several layers and inputs or outputs. It has limits.[14]

- Batch Normalization:

Standardization by lot is a technique that automatically standardises the inputs to a deep neural learning layer. Standardization by lots leads to substantial acceleration in the neural network training and a little regularisation impact to improve model performance in certain conditions. The layer normalises the entries, which means that they have both a mean of zero and a standard difference.

```
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Dense
from keras.layers import Flatten

from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.models import Sequential

from keras.layers import BatchNormalization
```

Fig 2 Code snippet of import modules

Data Preprocessing

The image data input must be preprocessed in order to feed Convolutionary Neural Networks with significant floating point tensors. Tensors are structures for data storage which can be considered multidimensional arrays[15]. A tensor with an image 64 × 64 with three channels should be measured (64, 64, 3). Let's examine the steps to get there. The data is currently saved on a disc as JPEG files, so see how we have arrived.

Model Creation

CNN models are commonly used for concentrated, pooling, drop-out and completely related layers. CNN works for the issues of picture categorization because the data represented as grid structures perform better. The dropout layer is used to disable portion of the neurons during training to minimise overfitting of the model. The model will then be compiled using the Adam Optimizer. We utilise the method 'add()' to add layers to our model.[13]

Our initial two layers include Conv2D layers. These are convolution layers that handle as input pictures with our two-dimensional matrices.

In each layer there are 64 nodes in the first and 32 in the second. This number might be changed higher or lower depending on the size. 64 and 32 work well in our experience and we're going to stay with them for the moment.

The size of the kernel is the filter matrix for our convolution. We will have a kernel size 3x3 filter matrix. Go back to the introduction and the first picture for a refresh. Activation is termed the layer activation function. The ReLU is our activation function for our first two layers, or the Rectified Linear Activation. This activation function has shown successful in neural networks.

A form is also acceptable for our initial layer. As previously said, each input picture has a shape that is 28,28,1, and 1 means grey scale photos.

Between Conv2D layers and dense layer exists a 'Flatten' layer. Flatten is used for connecting dense and convolutionary layers.

Our output layer is 'dense' with the type of layer used. Dense is a typical layer type used in many contexts in neural networks.

We shall have 10 nodes in our output layer, one for every imaginable result (0-9).

The activation is Softmax. Softmax lowers the output to a number, which may be interpreted as probabilities. On the basis of these information the model will provide a forecast.

 DOI : 10.35291/2454-9150.2022.0400

The algorithm then provides a forecast based on which alternative is most likely to succeed.

Compiling the model

Our model is the next phase. The model has three parameters to be built: optimizer, loss and metrics. The pace of learning is regulated by the optimizer. We are going to utilise the name 'adam' as an optmizer. Adam is mostly a good optimizer to use. The adam optimizer changes the learning rate throughout the training.

The study rate checks how quickly the optimum weight of the model is computed. A slower pace can lead to more accurate weights, but it takes more time to calculate weights.

Our loss function was 'comprehensive crossenterpy.' This is the most prevalent way of classifying. If the score is lower, the model works better.

Putting the model together

Use the generator to fit the data model. The fit generator approach is utilised, which is the same as for data generators. As the data is created indefinitely, before announcing an epoch, the Keras model needs to know how many samples to take from the generator. His first argument is a Python generator which provides infinite batches of inputs and targets. This is the purpose of the steps per epoch parameter.

Code:-

```
model.fit_generator(training_set,
                    steps_per_epoch = 10,
                    epochs = 100,
                    validation_data = val_set,
                    validation_steps = 100,
                    callbacks=callback_list
                   )
```

Fig 3 code

Now finally below is the illustration of our results which we got from the system of which we have shared some screenshots

Below are some Screenshots of the results:

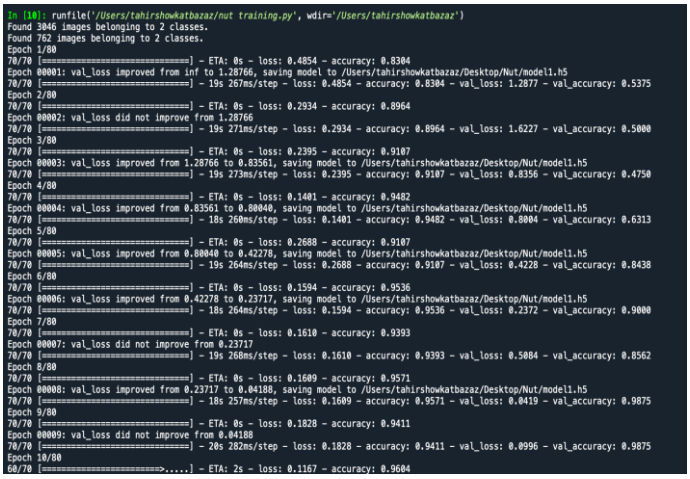


Fig 4 Model training and evaluation

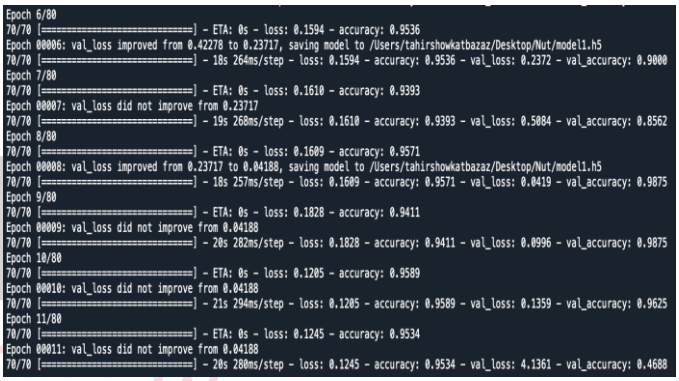


Fig 4 Model Saved

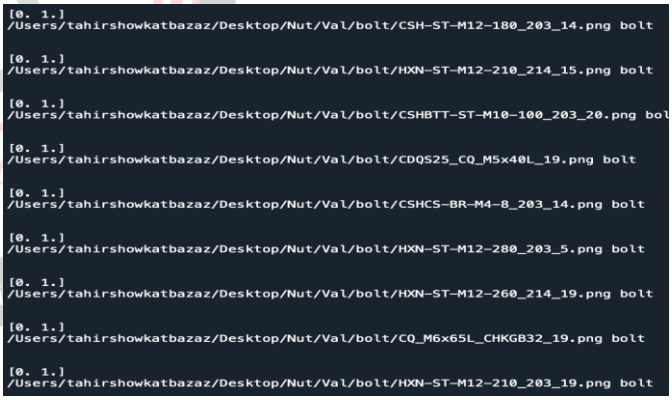


FIG 5 FINAL TEST OUTPUT

As u can see in fig 5 the final test output, the system is efficiently able to recognize the nuts and bolts for the purpose of separation which is our main goal.

V CONCLUSION

For things to be automatically and free of human effort, it is important to be able to recognise and categorize nuts and bolts correctly. There is a collection of techniques and classifiers for the recognition of photographs. We employed picture processing and CNN techniques in our research for this purpose. We used Spyder software to construct the main component analysis and an artificial neural network. For the same purpose the optimization

 DOI : 10.35291/2454-9150.2022.0400

technology needs less iterations, while the convergence rate is faster and the precision is higher, than the Artificial Neural Network and other upgraded techniques procedures. Curve numbers are used to get characteristics for object recognition in terms of perimeter radius. We imported all the modules that are needed to train our model and process the image data input in order to feed Convolutionary Neural Networks with significant floating point tensors and subsequently create and compile the model accordingly, and then finally put the model together (using fit generater approach) for work. Our technique is more effective and suited for real-time recognition systems composed on previous research as iteration time, belt transport speed and accuracy is enhanced.

ACKNOWLEDGMENT

. The authors thank the members of faculty and staff of the Department of mechanical engineering, RIMT university, mandi, gobindgarh, India for their immense support and assistance in carrying out this research work.

REFERENCES

[1] Akbar H. and Prabuwono A. S., "The design and development of automated visual inspection system for press part sorting," in Proc. International Conference on Computer Science and Information Technology (ICCSIT'08), 2008, pp. 683-686.

[2] Thomas G. Gunn. The mechanization of design and manufacturing. Scientific American, 247(3):114–131, 1982.

[3] David G. Ullman. The mechanical design process, 2010

[4] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In Proceedings Shape Modeling Applications, 2004., pages 167–178, June 2004

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, vol. 25, no. 2, Dec. 2012

[6] P. Felzenszwalb, D. Mcallester, D. Ramanan. "A discriminatively trained, multiscale, deformable part model," in IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[7] Jing Bai, Shuming Gao, Weihua Tang, Yusheng Liu, and Song Guo. Design reuse oriented partial retrieval of cad models. Computer-Aided Design, 42(12):1069 – 1084, 2010.

[8] Tensorflow documentation, 2018.

[9] Keras documentation, 2018.

[10] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[11] J. Redmon, S. Divvala, R. Girshick, et al., "You only look once: unified, real-time object detection," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.

[12] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[13] P. Felzenszwalb, D. Mcallester, D. Ramanan. "A discriminatively trained, multiscale, deformable part model," in IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[14] Antonio Cardone, Satyandra K. Gupta, and Mukul Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. J. Comput. Inf. Sci. Eng., 3:109–118, 2003.

[15] An multi-view cnn (mvcnn) implementation with tensorflow., 2015.

 DOI : 10.35291/2454-9150.2022.0400