Classification and Evaluation of Free-Hand Sketches Using Image Processing and Deep Learning Techniques

Md. Afzalur Rahaman *afzalurrahaman@yahoo.com*
Computer Science and Engineering
Hamdard University Bangladesh
Munshiganj-1510, Bangladesh

Fahima Hossain *minda.fahima25@gmail.com*
Computer Science and Engineering
Hamdard University Bangladesh
Munshiganj-1510, Bangladesh

Hasan Mahmud *hasanmahmud4277@gmail.com*
Computer Science and Engineering
Hamdard University Bangladesh
Munshiganj-1510, Bangladesh

Mahdi Hassan Sabbir *mahdihassan1998@gmail.com*
Computer Science and Engineering
Hamdard University Bangladesh
Munshiganj-1510, Bangladesh

Md. Masud Hasan *m.hasan61999@gmail.com*
Computer Science and Engineering
Hamdard University Bangladesh
Munshiganj-1510, Bangladesh

Abstract

Evaluation is a crucial issue in a learning system. Instructors frequently assign a collection of questions, which students must respond to in the script, in order to evaluate their performance. An answer is most often composed of text, equations, and figures. The sketched figures must be recognized and rated according to their actual appearance. With the advancement of computer vision, several methods have been developed for recognizing and grading handwritten text accurately. To ensure a fair automatic evaluation system, we must develop a system that can grade text and images simultaneously. Due to the complex structure of images, we need to extract important features in the image, unlike traditional text grading methods. The major focus of this research work is mostly on the freehand sketch phase, therefore developing a CNN model, that can classify and assign a grade to a given image automatically. The model is trained with a multi-labeled dataset where images are graded and labeled by the expert human evaluator. This dataset needed to undergo some preprocessing steps before being fed by the proposed CNN model.

Keywords: Deep Learning, Freehand Sketch Evaluation, Image Processing, Multi-labeled CNN.

1. INTRODUCTION

Hand-drawn sketches are used to visualize the most intuitive part of a particular function, organ, or architecture. They have been utilized as a fundamental tool for conveying feelings, thoughts, judgments, and opinions in education life, history, and human relations (Xu et al, 2023). Sketches are frequently images in binary or grayscale. They exhibit highly abstract and large intra-class deformations as a result of the incomplete outline caused by the user's drawing process's incomplete pause and discontinuity (Sert et al, 2017). Then, emphasize real-world object characteristics while ignoring features that are either less important or more difficult to draw. They are also used by children before writing and transcend language barriers. Sketches can be of other forms such as professional sketches, forensic sketches, cartoons, technical drawings, and oil paintings. They require no training and no special equipment. However, the sketch recognition problem differs from traditional photographic image classification due to its visual complexity. People's concepts of an object match, but their interpretations vary significantly. Sketches have been widely studied in computer vision and pattern recognition, computer graphics, human-computer interaction, robotics, and cognitive science. They are typically drawn on touch-sensitive devices like tablets, computers, and phones, which lack color and texture information (Lu et al, 2017). With the advancement of technology, people can now draw sketches and handwrite in digital form using graphic tablets and other touch-screen display and input devices (Fan et al, 2020). As a result, research on freehand sketches is expanding rapidly, including sketch recognition, image retrieval based on freehand sketches, and 3D model retrieval based on freehand sketches. Understanding and recognizing the content would make it more searchable and retrievable (Hayat et al, 2019). The most important step, however, is to comprehend and recognize the semantics of these sketches (Nanniet et al, 2017).

For the last four decades, computers have been used for taking exams and evaluating scripts homogeneously. In a pedagogy system, instructors frequently take exams, evaluate them, and provide feedback individually for a particular course. Unfortunately, this significantly increases the workload for teachers. Responding to student papers and rigorously checking them is a burden for many teachers, and this pressure increases linearly as the number of students increases. Therefore, developing an automated system can help to reduce the cost of checking in a significant way and facilitates students to get early feedback. Freehand sketch recognition is the ability of a computer to interpret an image from a script or from other sources. The script is scanned optically to trace the most plausible position of an image. The most challenging part of this work is parallel checking the relevance of a figure corresponding to the instructor's assignment as well as assigning a grade as per its correctness level simultaneously. The development of freehand sketch mostly started with Principle Component Analyzer (PCA) and Artificial Neural Network (ANN) although a satisfactory performance level wasn't achieved well. After the revaluation of Deep Learning (DL), algorithms such as CNN's architectures LeNet and AlexNet are capable of generating more distinctive features from sketch images and can leverage the performance for sketch classification or recognition as compared to the use of hand-crafted features. These techniques have outperformed traditional object recognition algorithms on multiple large-scale datasets, resulting in higher accuracy. In contrast, a small amount of research has been done on freehand sketch recognition and grading on a custom dataset for automatic evaluation purposes, and no model has been developed to be a replica of an expert human grader. Till now much research is being done in this sector and researchers investigating machine scores to move forward in their drive with the aim of improving the accuracy and effectiveness of the system.

Developing a multilevel classification and grading system of hand sketches can make several contributions to the field of computer vision and image processing. Some of these contributions are:

- Development of a multilevel classification and grading system that can reduce the need for manual grading and classification of hand sketches, which can save significant time and effort in applications where large numbers of sketches need to be evaluated.
- A custom CNN model was constructed to enhance the accuracy of the system by accounting for various levels of abstraction and complexity present in the hand sketches.

The proposed system is intended to be an aid in grading a large corpus of student answers, reducing the teacher's load, and thereby encouraging them to give students more free-hand tasks. The rest of the paper is organized as follows: Section 2 presents the relevant works on free-hand sketch recognition systems. Section 3 describes the architecture of the proposed system. Sections 4 and 5 are about dataset and model development. Section 6 presents the performance analysis. Section 7 concludes with a conclusion and future scope.

2. LITERATURE REVIEW

Qunjing Ji (2021) proposed a neural network, DSCN (Depthwise Separable Convolutions Net), for hand-drawn sketch recognition based on depth-separable convolution. Because the information from the sketch stroke sequence has a strong ability for feature extraction and nonlinear expression, the features are extracted in the order of arrangement of the original stroke sketch. To improve recognition ability, a time series relationship is built. The neural network is then trained and tested using the output features. The neural network is primarily made up of an encoder-decoder structure, with the encoder weighing the parameter scale, running speed, and recognition accuracy of the entire network. The random gradient descent algorithm is used as the parameter optimizer in network training; the learning rate is 0.001, the momentum is 0.9, and the training batch is 16. The last layer's activation function is sigmoid. The TU-Berlin dataset was used for the experiment.

An innovative and effective CNN-based methodology for identifying hand-free sketching was presented in (Hayat et al, 2019). To increase the model's capacity for generalization and discrimination, the image data is initially augmented. Then, to address overfitting difficulties, image augmentation is carried out and edge maps are generated utilizing various geometric transformations, such as flips and rotations. For transfer learning on an increasing quantity dataset, the ImageNet dataset, the authors in this study used three DCNN architectures: the Visual Geometric Group Network (VGGNet), Residual Networks (ResNet), and Inception-v3. This trained knowledge is then used to generate feature maps. The GAP (Global Average Pool) layer is used to compress these feature maps. The SoftMax classifier is then used to categorize and identify these circumstances. The TU-Berlin sketch dataset was used for the experiment.

In order to reduce the cost of training, EmelBoyaci et al (2017) devised an effective method for identifying hand-free sketching. The authors developed a feature-level fusion technique in which they utilized two ImageNet pre-trained CNN networks, such as AlexNet and VGG19, to extract abstract feature vectors from sketches. These features were then integrated using the concatenation operator and unified using the L2 norm. Dimensionality and time complexity was decreased in CNN architectures by using pooling and convolution layers. A radial basis function (RBF) is used as the kernel function in the SVM classification technique before these feature vectors are fed into it. The evaluation made use of the dataset (1,350 participants) gathered by Amazon Mechanical Turk.

To improve the recognition rate of a hand-drawn sketch, Lei Zhang (2021) founded a hand-drawn sketch recognition algorithm based on a dual-channel convolutional neural network. The input image is first smoothed, and the contour features are extracted from it. These features are then provided to CNN. In this work, CNN's fully connected layer performs feature fusion, and Softmax is used for classification. In this particular instance, CNN operates in two channels to access different data views (such as red, green, and blue channels of color images and stereo audio tracking). These two channels form local connections between pixel blocks and neurons, reducing the number of parameters in the deep web, the model's complexity, and the training speed. The concept of back propagation is also used here for updating weights and reducing the number of parameters used in neural networks.

A system for categorizing and assessing freehand sketches was designed by (Chandan C.G. et al, 2018). The sketches are scaled from [0, 255] to [0, 1.0], which is the most often used scaling scheme, to enhance the learning capacity of neural networks. Subsequently, utilizing DoG (Difference of Gaussian) and SIFT (Scale Invariant Feature Transform), significant characteristics are extracted from sketches using key points and descriptors, and these key points are then employed for grading the sketches. The ReLu activation function is then used to activate the output features as input in CNN's LeNet architecture for training and testing. The CNN score and SIFT-matching score are used to grade the sketches. Based on the test sketches, a match between the scoring sketches is discovered. Based on these key points and generated descriptors, a match between the scoring sketches and the test sketches is determined. The number of matches, the number of key points in the scoring sketch, and the number of key points in the testing sketch are used to determine the score between the test image and the scoring image. The final score represents the highest possible score.

Sketch classification is a challenging and complex task as classification relies on extracted features for training before making recognition. Handcrafted features were considered in traditional methods, but they cannot be used in modern methods due to their high dimensionality, which makes them computationally expensive (L. Nanniet al, 2017).

Most existing systems are mainly developed for hand-drawn sketch recognition only, and no automated system is yet developed for simultaneous recognition and evaluation of sketches in the educational scope with a high enough performance level to be a replica of a human grader. This indicates much more research work needs to be done. To address the issues, we presented a system, the architecture of which is briefly detailed in the next section.

3. ARCHITECTURE OF PROPOSED SYSTEM

For the work in this paper, we first formulate the problem and then describe the architecture of the proposed system with the background architecture of convolutional neural networks and the Adam optimizer.

3.1 ConvolutionalNeural Network (CNN)

CNN is a multilayer neural network architecture specifically designed to process two-dimensional data. The performance of CNN is significantly better than other methods. CNN is mostly utilized in image processing tasks like image classification and identification. CNN can examine an image as input data, extract various features from that data, and then make decisions based on those features. Convolution layers, pooling layers, and fully connected layers are some of the many building blocks used in the CNN technique to gradually extract various information from an input image. Convolutional layers convert the input image into a stack of filtered images so that kernels or filters can be used to extract features. The purpose of the activation layer is to combine the results from various layers for the next convolution process. Then, to enhance computational performance, the dimensions of the features from the convolutions are down-sampled using polling layers. The fully connected layer was eventually filled with the outputs of the convolutional and pooling layers. CNN is widely used in object detection, object recognition, and pattern classification.

3.2 Adam Optimizer

Adaptive moment estimation is an algorithm for optimizing the techniques for gradient descent. The method is efficient when a problem involves a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the ‘gradient descent with momentum algorithm and the ‘RMSP’ algorithm. Instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam also makes use of the average of the second moments of the gradients. Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters β1 and β2 control the decay rates of these moving averages. The initial value of the moving averages β1 and β2 values are close to 1.0. We compute the decaying averages of past and past squared gradients $m_t$and $v_t$respectively as follows:

$$m_t = \beta_{1m_{t-1}} + (1-\beta_1)g_t$$
$$v_t = \beta_{2v_{t-1}} + (1-\beta_2)g_t^2$$

Here, $m_t$and $v_t$are estimates of the first moment and second moment of the gradient respectively. As they are initialized as vectors of 0'th, they are biased towards zero, especially during the initial time step and when decay rate is small ($\beta_1$and β_2 are close to 1). This incident is counteracted by computing bias correlated first and second moment estimates as follows:

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

To update the parameters, as we have seen in Adadelta and RMSprop, this yields the Adam update rule as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \widehat{m_t}$$

Empirically, Adam initialized the value of 0.9 for β1, 0.999 for β2, and 10-8 for ε.

We have partitioned the research work into three phases: (i) preprocessing (ii) training the model and (iii) testing a new sketch. The block diagram of the developed system is shown in Figure 1 and the functionalities of different units are described as follows:

3.3 Preprocessing

To train our model, we need to convert and represent our collected data in such a way that the neural network model for our project can understand the data and train our machine appropriately. The data preprocessing involves several individual steps like augmentation, filtering, resizing, and normalization. These steps are going to be articulated in the forthcoming paragraphs.

3.3.1 Data Augmentation

Data augmentation is a group of techniques that improve the size and quality of training datasets so that they support a stronger deep-learning model. Modern machine learning models often demand enormous volumes of high-quality annotated data, thus, it is necessary to assure optimal performance. To prevent the model from being unduly biased towards classifying examples as the majority class type, it is also used to re-sample imbalanced class distributions. It often scales, rotates, and makes a wide range of modifications so that the neural network is granted a large number of different kinds. The neural system becomes less willing to learn unwanted features as a result. The initial number of images obtained was relatively low. We performed data augmentation by applying geometric and morphological transformations to each sketch, such as dilation and distortion, to increase the number of sketches per category for classification.

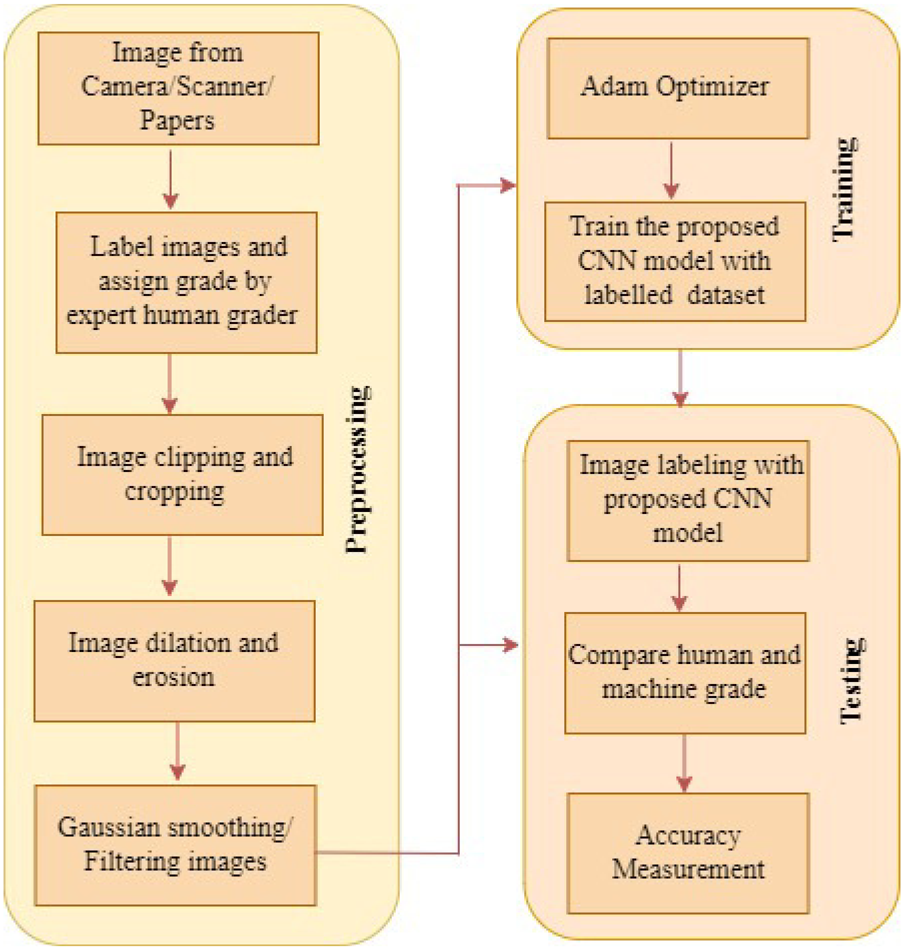


Image from Camera/Scanner/ Papers
Label images and assign grade by expert human grader
Image clipping and cropping
Image dilation and erosion
Gaussian smoothing/ Filtering images
Preprocessing
Adam Optimizer
Train the proposed CNN model with labelled dataset
Training
Image labeling with proposed CNN model
Compare human and machine grade
Accuracy Measurement
Testing

FIGURE 1: Flow chart of the proposed method.

3.3.2 Dilation

Dilation is the process of enlarging or constricting an image. To do this, extra pixels are added to an image's object boundaries. The size and shape of the structuring element used to process the image determine how many pixels are added to the image.

3.3.3 Distortion

Image distortion causes an image's straight lines to appear to be twisted or curved in an unnatural way, leading to various types of distortion, such as waveform, pincushion, and barrel. The quality of the image might be considerably hampered by distortion. Data augmentation produces several fresh sketches for each original sketch. The dataset sketches are dilated, which we feel improves detail preservation, to reduce the impact of detail loss during the training process.

3.3.4 Filtering

There may be a lot of noise in images captured in low light. Therefore, in this work, the noise is minimized using Gaussian smoothing. It might soften the image. It is a low-pass filter used to blur certain areas of an image and reduce noise. To get the desired result, the filter is constructed as an odd-sized symmetric kernel and is applied to each pixel within the region of interest.

3.3.5 Resizing

128 x 128 x 3 pixel input sketches have been taken into consideration in this study.

3.3.5 Normalization

Normalization is a technique used in image processing to change the range of pixel intensity values. This is a crucial stage where we confirm the distribution of the input data. As a result, network training converges more quickly than usual. Its typical function is to transform an input image into a set of pixel values that are more normal or familiar to the senses. Normalization is the term used to describe scaling data to a 0–1 range. In this work, we have executed a function that normalizes an input image (grayscale or RGB). Additionally, we scale down our designs from [0, 255] to [0, 1.0], the most common scaling system. By doing this, for instance, those extremely dark images become more discernible. Since we are normalizing an RGB (3-channel) image, we must use the same standards to normalize each channel. We applied the normalization of each image using the following equation:

$$Image_{new} = \frac{Image - image_{min}}{image_{max} - image_{min}}$$

3.4 Training Model

In this step, the model is trained using the features retrieved from the sketch dataset. Each image is then labeled with the category and grade, which are assigned by humans. For better accuracy, we assigned the grade of each image to three expert human graders.

3.5 Testing Sketch

Once the model is trained well with a satisfactory level of accuracy compared to human graders, the model at this stage is used to assign category and grade to an unknown sketch.

4. DATASET DEVELOPMENT

To conduct our research on the stated problem, the most curtail task was to collect the data set. We had to collect our own dataset, since this type of data was not available. At the very beginning of our research, we have arranged a freehand sketch drawing program at two different elementary schools to collect our desired image data. The students were limited to drawing images within seven categories, but they were free about image quality since we need not only good images but also average-quality images for better accuracy. Finally, we have developed a dataset with 836 hand-drawn images with seven classes and two grades. Figure 2 displays a sample of seven different classes of images. The presence of ambiguously drawn sketches with identities that are challenging to determine even for fellow humans is one of the main flaws. They are unable to precisely describe what the object is. In order to resolve this issue, this system will implement a human-evaluation-based technique and identify a subset including 7 non-ambiguous object categories in which the incredibly subpar sketches were excluded by a professional in this field. A labeled dataset was produced by this procedure. The same expert grades the remaining sketches. The sketches from this labeled dataset will be used in our experiments to evaluate how well the sketch recognition and grading system performs. In table 1 the total number of data with their types and the distribution for the train and test sets have been shown.

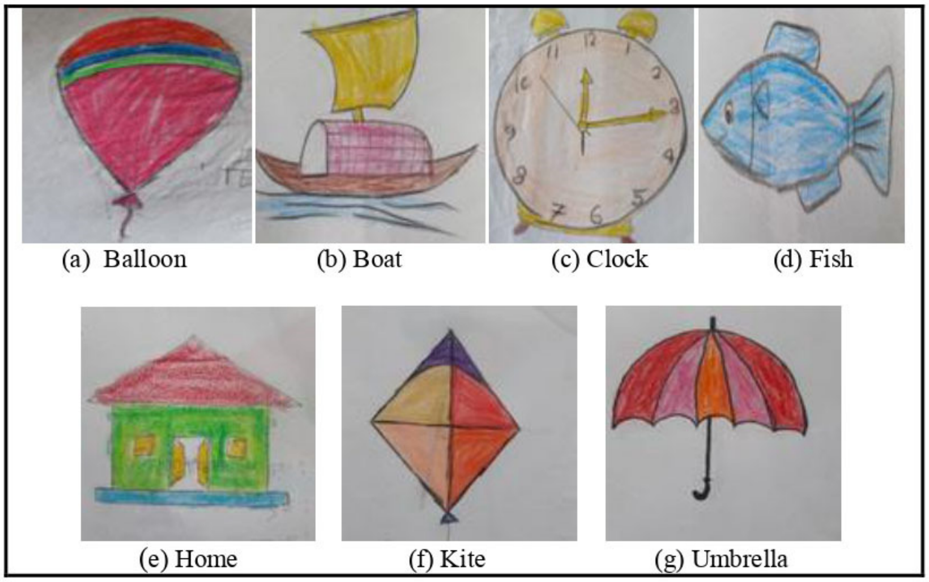


FIGURE 2: Freehand sketches of several object groups.

Category	Number of images	Train (~80%)	Validation (~20%)
Balloon	83	63	20
Boat	130	98	32
Clock	104	74	26
Fish	140	105	35
Home	140	105	35
Kite	90	68	22
Umbrella	149	112	37
Total	836	629	207

TABLE 1: Dataset used for the proposed CNN model.

5. MODEL DEVELOPMENT

After the processes of data collection and preprocessing stages of our project, we moved on to the next level of our research. We have to analyze our data to find necessary association among the dataset. In our project we applied deep learning approaches to make our machine enable to find necessary association about the dataset that we collected. CNN is the most useful and popular tool among all the existing algorithms to analyze image data. We have developed a multi-labeled CNN model as the proposed system. Applications for CNNs include everything from computer vision to natural language processing. A convolutional neural network utilizes a back propagation algorithm to learn the spatial hierarchy of data automatically and adaptively. It has several layers, including a convolution layer, a pooling layer, a flattened layer, a dropout layer, a batch normalization layer, and a fully connected layer. The architecture of CNN used in this work is illustrated in the figure 3.

5.1 Foundation of Developed CNN model

5.1.1 Convolutional Layer

A set of kernels known as convolutional layers are used to execute the convolution operation (matrix) on an input picture in order to create a feature map. In order to build more elaborate models to extract nuanced information from images, convolutional layers can be stacked. A non-linear activation function is used to process the outputs of linear processes like convolution. The most popular non-linear activation function today is the rectified linear unit (ReLU) as follows:

$$f(x)_{ReLU} = \max(0, x)$$

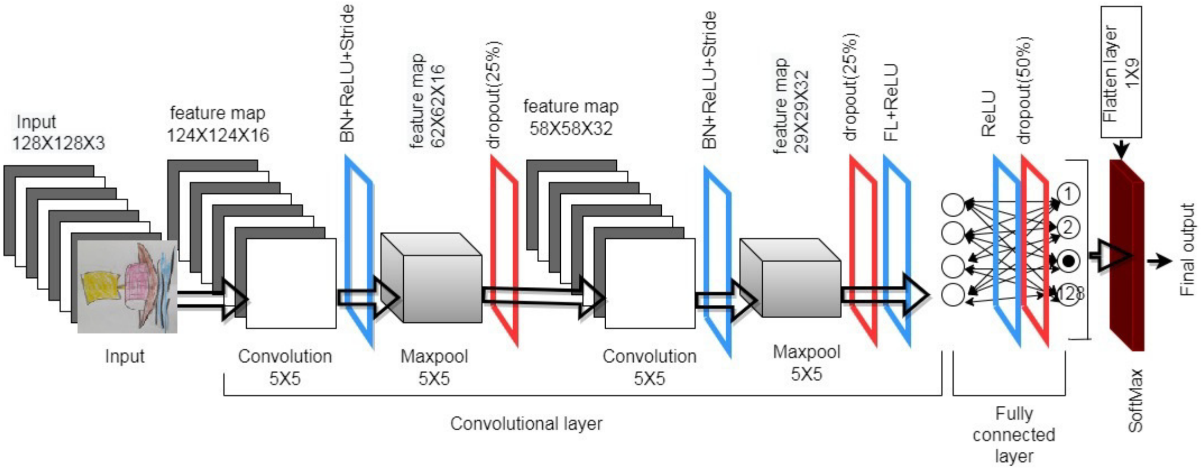


FIGURE 3: Overview of proposed CNN.

5.1.1 Polling Layer

The output sample from the convolutional network before it is down-sampled using a pooling layer. It contributes to demanding less memory and less processing power. Pooling speeds up training and also contributes to a decrease in the number of parameters. Max pooling and average pooling are the two primary types of pooling. While average pooling uses the average value from each feature map, max pooling uses the maximum value.

5.1.2 Dropout Layer

The regularization method used to stop over fitting in the model is called dropouts. In every training step, we randomly remove a portion of a layer's input units, which makes it much more difficult for the network to recognize those erroneous patterns in the training data. Instead, it must look for wide, all-encompassing patterns, whose weight patterns typically exhibit more robust patterns. A group of smaller networks will now make the predictions rather than just one large network. To help the model learn more effectively, this is done.

5.1.3 Flatten Layer

Convolution layer to full connected layer transitions frequently use the flatten layer to reduce the multidimensional input to one dimension.

5.1.4 Batch Normalization Layer

When training very deep neural networks, a technique called batch normalization is used to equalize the inputs to each layer for each mini-batch normalization layer for its inputs. It uses a transformation to keep the output standard deviation and mean close to one and zero, respectively. The learning process is stabilized as a result, and the quantity of training epochs needed to train deep networks is drastically decreased.

5.1.5 Fully Connected Layer

Data eventually travel across the network and arrive at the top layer, also known as the fully connected layer. Fully connected layers use the features obtained by earlier layers to predict outcomes. The final output category is determined by the fully connected layer, which also provides a set of output data. FC layers are present near the end of CNN architectures; from the set of the final convolutional or polling layer, using a probability distribution method, it chooses the output with the highest probability value, which can be defined as follows:

$$\sigma(x_i) = \frac{e^{xi}}{\sum_{j=1}^{K} e^{xj}} \; for \; i = 1, 2, \ldots\ldots K$$

Where,
σ = Softmax function,
x_i = Input set,
K=Number of classes,
e^{xi} = Standard exponential function for input vector

6. ARCHITECTURE OF DEVELOPED CNN MODEL

The model is developed with three convolutional layers. The first layer has 16 convolution filters, where each kernel is of size 5 x 5, applied to the ReLU activation function, followed by a 2 x 2 max-pool layer in both the x and y directions with a dropout of 0.25. The second convolutional layer has 32 convolution filters, where each kernel is of size 5 x 5, applied to the ReLU activation function, followed by a 2 x 2 max-pool layer in both the x and y directions and a dropout of 0.25. The third convolutional layer has 64 convolution filters, where each kernel is of size 5 x 5, applied to the ReLU activation function, followed by a 2 x 2 max-pool layer in both the x and y directions, and a dropout of 0.25 is used. The model is developed with three fully connected layers. The first fully connected layer consists of 512 neurons, ReLU activation function is applied and a dropout of 0.50 is used. The second fully connected layer consists of 256 neurons, the ReLU activation function is applied and a dropout of 0.50 is used. The third fully connected layer consists of 128 neurons, ReLU activation function is applied and a dropout of 0.50 is used.

All the convolutional layers use the ReLU activation function to provide nonlinearity to the linear convolutional layer and decrease the issue of over fitting. ReLU is far more computationally efficient than the sigmoid and tanh functions since it only activates a small subset of neurons. A data-dependent hyper-parameter used for tuning when training a neural network with mini-batch Stochastic Gradient Descent (SGD) is the batch size parameter.

Input size 128x128x3, Batch size: 32, Epochs: 50			
Layers		**Output size**	**Configuration**
Convolution 1	Convolution	124x124x16	Filters: 16, kernel size: 5x5, Batch Normalization, activation: ReLU
	Polling	62x62x16	2x2 max pooling
	Dropout	62x62x16	Dropout rate: 0.25
Convolution 2	Convolution	58x58x32	Filters: 32, kernel size: 5x5, Batch Normalization, activation: ReLU
	Polling	29x29x32	2x2 max pooling
	Dropout	29x29x32	Dropout rate: 0.25
Convolution 3	Convolution	25x25x64	Filters: 64, kernel size: 5x5, Batch Normalization, activation: ReLU
	Polling	12x12x64	2x2 max pooling
	Dropout	12x12x64	Dropout rate: 0.25
Flatten (1x1) 9216			
Fully connected layer (1x1), 512 Neurons, ReLU, Dropout rate: 0.50			
Fully connected layer (1x1), 256 Neurons, ReLU, Dropout rate: 0.50			
Fully connected layer(1x1), 128 Neurons, ReLU, Dropout rate: 0.50			
SoftMax layer: 10 labels (category: 7, grade: 3)			

TABLE 2: Set of configurations in the CNN model.

The simplest way to find a hyper-parameter pair that causes the network to converge is to grid search over the learning rate and batch size. The optimal batch size for neural network learning is 32. CNN's set of configurations is represented visually in Table 2. After the training, the model is tested as a classifier using a test set of previously unseen images.

7. PERFORMANCE ANALYSIS

We assigned the training and testing datasets a ratio of 80:20. This ratio was discovered through trial and error. The accuracy for both training and validation is higher with this ratio.

7.1 Training and Validation Accuracy

Figure 4 shows how the model's performance varies throughout the course of the epochs in terms of training accuracy and testing accuracy. In both the training and testing processes, the classification accuracy would rise with the passing of epochs. This indicates that the model is growing. We used a total of 50 epochs and end of the compilation we achieved almost 83% training and close to 100% accuracy. Figure 5 illustrates how the model's performance varies in terms of training loss and validation loss throughout the duration of the epochs.

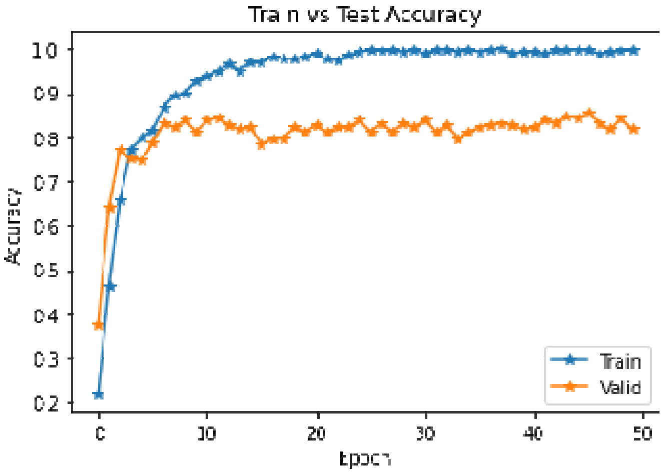


FIGURE 4: Training Accuracy and testing accuracy vs. epoch.

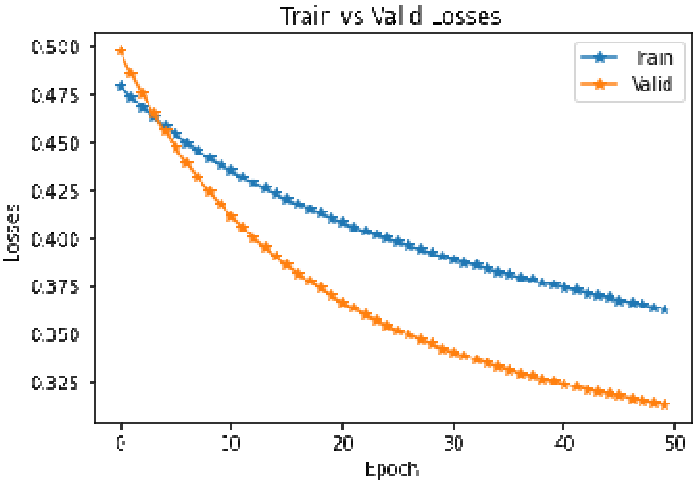


FIGURE 5: Training loss and validation loss vs. epoch.

With each passing period in a conventional training and testing procedure, the classification loss would diminish. As the number of epochs rose, training and validation losses decreased.

Category	Number of images	Machine accurately labeled images	Human graded as best	Machine accurately graded as best	Human graded as good	Machine accurately graded as good	Labeling accuracy (%)	Grading accuracy (%)
Balloon	20	17	10	10	10	7	85%	85%
Boat	19	14	9	8	10	8	74%	84%
Clock	19	17	10	10	9	7	89%	89%
Fish	36	35	18	16	18	15	97%	86%

Home	39	34	20	18	19	16	87%	87%
Kite	36	32	18	18	18	16	89%	94%
Umbrella	38	34	19	18	19	15	89%	86%
Total	207	183	104	98	103	84	88%	88%

TABLE 3: Performance analysis of proposed CNN model for 207 test images.

We have applied a number of metrics and curves to assess our suggested model. We used an input with a size of 128x128x3, a batch size of 32, and 50 epochs. The validation accuracy is calculated, and the network weights are stored after each epoch. The set of weights with the highest validation accuracy is utilized to calculate the test accuracy following the training epochs. Table 3 illustrates the performance of the developed CNN model. The performance of the model is measured on 207 different test images. In this table, the classification and evaluation accuracy are measured by comparing three valuations of an actual human being. For simplicity, we just assigned only two grades for a particular image; although the grading level could be customized if required. Although the distribution is the same for different categories, the number of datasets has a focused variation. This is because we didn't get fair distributed data while the dataset collection. In this regard, we had to balance the proportion with different data augmentation techniques. The final two attributes in the table show the labeling accuracy and grading accuracy of the developed CNN model. They are defined as follows:

$$\text{Labeling accuracy} = \frac{Number\ of\ images\ acurately\ labeled\ by\ Model\ of\ a\ category}{Number\ of\ imaged\ labeled\ by\ Human\ of\ a\ category}$$

$$\text{Grading accuracy} = \frac{Number\ of\ images\ acurately\ graded\ by\ Model\ of\ a\ category}{Number\ of\ imaged\ graded\ by\ Human\ of\ a\ category}$$

7.2 Confusion Matrix

A table called a confusion matrix is frequently utilized in classification problems. In this table, occurrences in a true class are shown as rows, and occurrences in a predicted class are shown as columns. The elements of the confusion matrix are defined as True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

7.2.1 True Positive (TP)

The number of instances that belong to the target class and are appropriately identified as members of the target class is represented by TP. In this experiment, if the proposed system gives the "best" grade for which the human grade is "best," then the result is a true positive, which is shown as follows:

$$\text{TP} = \frac{Number\ of\ images\ predicted\ as\ Best\ by\ machine}{Number\ of\ images\ assigned\ as\ Best\ by\ human}$$

Figure 6 shows, 98 images are truly categorized as positive out of 117 images. This indicates the True Positive Rate (TPR) is 84%. The same case is for Figure 7; at the first position, we can see that, out of 20 images of a balloon, 17 were accurately recognized.

7.2.2True Negative (TN)

The number of instances (TN) that are appropriately identified as not being members of the target class but do not belong to it In this experiment, if the proposed system gives a "good" grade for which the human grade is "good," then the result is a true negative, as shown as follows:

$$\text{TN}= \frac{Number\ of\ images\ predicted\ as\ Good\ by\ machine}{Number\ of\ images\ assigned\ as\ Good\ by\ human}$$

Figure 6 shows, 84 images are truly categorized negative out of 90, which indicates the True Negative Rate is 94%.

7.2.3False Positive (FP)

The number of instances that are incorrectly identified as the target class but do not belong to it is represented by FP. In this experiment, if the proposed system gives the “best” grade, for which the human grade is "good”, then the result is a true negative, as shown as follows:

$$\text{FP}= \frac{Number\ of\ images\ falsely\ assigned\ as\ Best}{Number\ of\ images\ assigned\ as\ Best}$$

Figure 6 shows 6 images are misgraded as “best” by the model out of 104 images. Therefore, the False Positive Rate (FPR) is 5%.

7.2.4FalseNegative (FN)

FN is the number of instances that are mistakenly labeled as belonging to a different class but actually belong to the target class. In this experiment, if the proposed system gives a “good” grade for which the human grade is "best”, then the result is a false negative and is shown as follows:

$$\text{FN}= \frac{Number\ of\ images\ falsely\ assigned\ as\ good}{Number\ of\ images\ assigned\ as\ Good\ by\ human}$$

Figure 6 shows 19 images are misgraded as “good” by the model out of 103 images. This indicates the False Positive Rate (FPR) is 18%, which is a bit higher compared to others. Finally, we can measure the accuracy, which is 88% from the confusion matrix equation as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

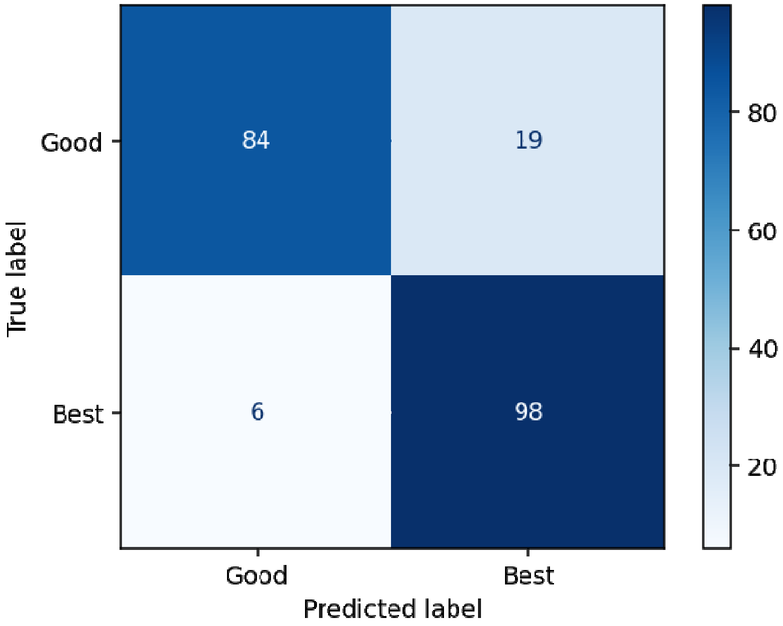
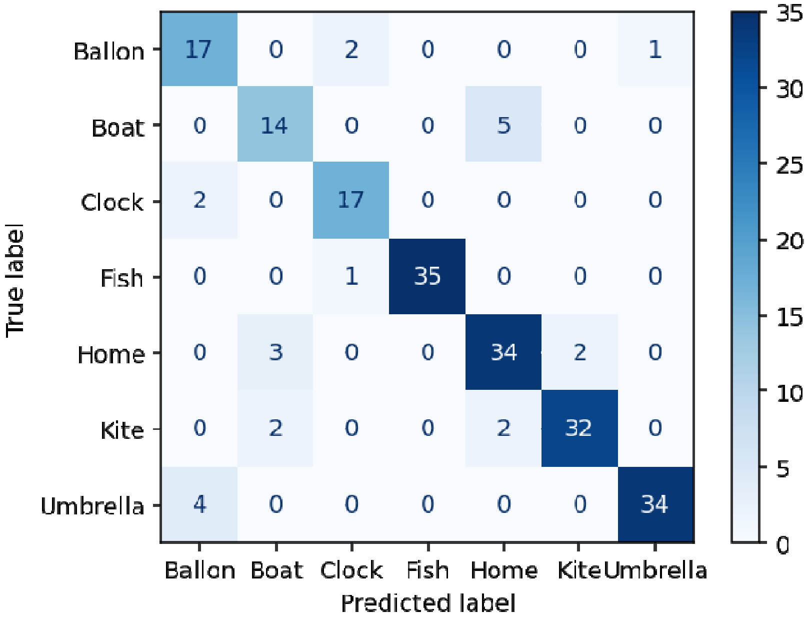


FIGURE 6: Confusion Matrix for freehand sketch grading.

Figure 7 represents the confusion matrix for the Sketch recognition performance of the proposed model. Since we have 7 categories of images, after summing up the diagonal elements (which we accurately recognized), we have found 187 images out of 207 images. This indicates the freehand sketch recognition accuracy is 88%.



True label \ Predicted label	Ballon	Boat	Clock	Fish	Home	Kite	Umbrella
Ballon	17	0	2	0	0	0	1
Boat	0	14	0	0	5	0	0
Clock	2	0	17	0	0	0	0
Fish	0	0	1	35	0	0	0
Home	0	3	0	0	34	2	0
Kite	0	2	0	0	2	32	0
Umbrella	4	0	0	0	0	0	34

FIGURE 7: Confusion Matrix for freehand sketch recognition.

The model was is built using three convolutional layers which has helped the model learn increasingly abstract and high-level features in the data, while adding three fully connected layers has provided more opportunities for the model to learn complex nonlinear relationships between these features. The softmax layer at the end of the model has typically been used for classification tasks and can help produce a probability distribution over the possible classes. Additionally, data augmentation and normalization are two common techniques that have been used in data preprocessing to improve the performance and accuracy of the model. In this work, augmentation helped to increase the size of the training data as the original training data is limited. It has also reduced the risk of over fitting. Furthermore, normalization reduced the scale of the input features, which helps to reduce the gradient vanishing or exploding problem thus produced improved accuracy by reducing the effect of differences in the scale of input features.

8. CONCLUSION AND FUTURE WORK

Developing a full-proof automated answer script evaluation system is a massive task because of the complex structure of answer scripts, which include figures, equations, and text with a variety of lengths, shapes, and approaches to the solution. In addition, evaluating hand-drawn sketches of large size requires a complex network structure. Developing a full-proof automated answer script evaluation system can benefit several stakeholders, including educators allowing them to focus on other aspects of teaching and assessment, students providing instant feedback and identify areas where they need to improve and work on their weaknesses, educational institutions reducing the cost and time required for grading answer scripts, exam boards and testing agencies reducing the possibility of errors and inconsistencies and so on.In this paper, we developed a model to recognize the category and assign a grade to a hand-drawn sketch. The model is developed with a variety of possible approaches by adjusting parameters, deep layers, number of neurons, activation function, and layers. We tuned each parameter many times, and adjusted layers and nodes to discover the perfect model. We also analyzed the performance with the test set and achieved the nearest 88% accuracy for both image recognition and grading. With the integration of handwritten text grading into this model, a robust automated answer script system could be developed. We are working on the same with the aim of developing a model like an expert human grader, and we believe this will potentially free the instructions from monotonous script checking.

Here is some research questions related to the development of a full-proof automated answer script evaluation system:

1. What is the optimal balance between human grading and automated grading in an answer script evaluation system, and how can this balance be maintained to ensure accuracy and reliability?
2. What is the impact of using an automated evaluation system on student motivation, engagement, and learning outcomes, and how can these effects be optimized?
3. How can an automated evaluation system be adapted and customized to different types of assessments, including standardized tests, entrance exams, and classroom assessments, and what are the benefits and challenges of such customization?

9. ACKNOWLEDGEMENT

The authors are grateful to the students of Barhatua government primary school, Rajapur government primary school, and Daffodil international school. The authors are also especially thankful to Nushrat Jahan, Ismat Jahan, Sanjana Rahman Promi, and Rabiul Auwal for their immense contribution to dataset development.

10. REFERENCES

Alex Krizhevsky, Ilya Sutskever, & Geoffrey E. Hinton.(2017). ImageNet classification with deep Convolutional neural networks. Commun. ACM 60, 6, 84–90.

C. Chandan, M. Deepika, S. Suraksha and H. Mamatha. (2018). "Identification and Grading of Freehand Sketches Using Deep Learning Techniques", 2018 *International Conference on Advances in Computing, Communications and Informatics (ICACCI).*

E. Boyaci and M. Sert. (2017). "Feature-level fusion of deep convolutional neural networks for sketch recognition on smartphones", 2017 *IEEE International Conference on Consumer Electronics (ICCE).*

Ghosh, A. & Sufian, A. & Sultana, Farhana & Chakrabarti, Amlan & De, Debashis.(2020). Fundamental Concepts of Convolutional Neural Network.10.1007/978-3-030-32644-9_36.

Islam, Sheikh & Hasan, Md. Mahedi, Abdullah & Sohaib. (2018). Deep Learning based Early Detection and Grading of Diabetic Retinopathy Using Retinal Fundus Images.

L. Fan. (2020) "Face sketch recognition using deep learning", PhD, Cardiff University, United Kingdom.

L. Nanni, S. Ghidoni & S. Brahnam. (2017). "Handcrafted vs. non-handcrafted features for computer vision classification", *Pattern Recognition*, vol. 71, pp. 158-172.

L. Zhang. (2021) "Hand-drawn sketch recognition with a double-channel convolutional neural network", *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1.

M. A. Rahaman, M. Mahin, M. H. Ali & M. Hasanuzzaman. (2019) "BHCDR: Real-Time Bangla Handwritten Characters and Digits Recognition using Adopted Convolutional Neural Network," 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-6.

Mignot, R., & Peeters, G. (2019). An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task. Transactions of the International Society for Music Information Retrieval.

Mishra, R. K., Reddy, G. Y., & Pathak, H. (2021). The understanding of Deep Learning: A Comprehensive Review. Mathematical Problems in Engineering.

M. Sert and E. Boyacı. (2019). "Sketch recognition using transfer learning", *Multimedia Tools and Applications,* vol. 78, no. 12, pp. 17095-17112.

N. Singh & H. Sabrol. (2021). "Convolutional Neural Networks-An Extensive arena of Deep Learning. A Comprehensive Study", Archives of Computational Methods in Engineering, vol. 28, no. 7, pp. 4755-4780.

Park, Keumsun, Chae, Minah & Cho, Jae.(2021). Image Pre-Processing Method of Machine Learning for Edge Detection with Image Signal Processor Enhancement. Micromachines. 12. 73. 10.3390/mi12010073.

P. Xu, T. M. Hospedales, Q. Yin, Y. -Z. Song, T. Xiang and L. Wang. (2023). "Deep Learning for Free-Hand Sketch: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 285-312,doi: 10.1109/TPAMI.2022.3148853.

Q. Ji. (2021). "Research on Recognition Effect of DSCN Network Structure in Hand-Drawn Sketch", *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1-12.

Rahaman, M. A., & Mahmud, H. (2022). Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach. Transactions on Engineering and Computing Sciences, 10(4). https://doi.org/10.14738/tmlai.104.12831.

Rahaman, M.A. and Hoque, A.S.M.L. (2022) "An effective evaluation system to grade programming assignments automatically", Int. J. Learning Technology, Vol. 17, No. 3, pp.267–290.

Rahaman, M.A., Latiful Hoque, A.S.M. (2019). Automatic Evaluation of Programming Assignments Using Information Retrieval Techniques. Proceedings of International Conference on Computational Intelligence and Data Engineering. Lecture Notes on Data Engineering and Communications Technologies, vol 28. Springer, Singapore.

Sarvadevabhatla, Santosh R. K. & Babu, R. (2015). Freehand Sketch Recognition Using Deep Features.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for Deep Learning. Journal of Big Data, 6(1).https://doi.org/10.1186/s40537-019-0197-0.

S. Hayat, K. She, M. Mateen and Y. Yu.(2019). "Deep CNN-based Features for Hand-Drawn Sketch Recognition via Transfer Learning Approach", *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9.

W. Lu, E. Tran. (2017). "Free-hand Sketch Recognition Classification", Stanford University.
Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. Array, 16, 100258.https://doi.org/10.1016/j.array.2022.100258.

Y. Li (2015). Free-hand sketch recognition by multi-kernel feature learning, Comput. Vis. Image Understand.