Premier Critical Perspective | Vol. 5, Issue 2, May 2022 | 103-125

Artwork Classification and Recognition System based on Convolutional Neural Network

Tanni Dhoom[1], Taufique Sayeed[2]

Abstract

From ancient ages, artworks have been the object of research in artist identification. Expert art historians primarily handle this issue manually. But an automatic artist recognition system using artwork is compulsory to lower the error percentage, and only a few progressive efforts are undertaken in this field, especially on Bangladeshi Artists. Our Convolutional Neural Network (CNN) model aims to determine the painter of a painting with a satisfactory accuracy standard. There are 450 paintings from 6 well-known Bangladeshi artists comprised in our novel dataset. Two different convolution kernels are used in model design, Model-1 has 3 X 3 convolutional kernels, and Model-2 has 5 X 5 kernel size. Our models achieve significantly higher classification accuracy as 87% for Model-1 and 89% for Model-2. Our result evaluation demonstrates that CNNs is not merely a robust learning tool for artist identification but also effective in predicting unique styles of an artisan.

Keywords: Deep learning, Convolutional Neural Network, Art, Bangladeshi artist.

1. Introduction:

Museums have started to integrate their vast and valuable art collections into digital libraries as image data acquisition technology has evolved over the last few decades. The objective of the technology developer is to facilitate digital picture acquisition, storage, and database search, which brings together image analysis researchers and art historians (Johnson, Hendriks, Berezhnoy, Brevdo, Hughes, Daubechies, Postma & Wang, 2008). Digital image conversion and analysis aid in the organization of artworks in databases, comparing artist styles (Saleh, Abe, Arora & Elgammal, 2016), and the technically accurate identification of artists. It is possible to compare and explain the influence of

[1]Tanni Dhoom
Assistant Professor
Department of Computer Science & Engineering, Premier University, Chittagong
Email: tanni.dhoom@puc.ac.bd

[2]Prof. Dr. Taufique Sayeed
Dean, Faculty of Science
Chairman, Department of Computer Science & Engineering, Premier University, Chittagong
Email: taufique.sayeed@gmail.com

different art styles on other artists, for example. With various image processing technologies, machine learning, and deep learning techniques, digital representation can aid the art expert in identifying the artwork (Johnson et al., 2008).

Bangladesh has a perennial history of its art, culture, and heritage, which originated more than two thousand years ago and is practiced even to this date. Renowned Bangladeshi artists greatly influence creating diversity in art and culture. Zainul Abedin is the pioneer of modern painting in Bangladesh. His sketches of the famine of 1943 brought the international popularity of Bangladeshi painting. Among the descendants, S.M. Sultan, Kamrul Hassan, Shahabuddin Ahmed, Rafiqun Nabi, Qayyum Chowdhury, and many more have made impressive contributions to Bangladeshi art and also represented Bangladesh on the international stage. Paintings can be divided into two forms: Eastern and Western styles. The former painting styles are generated from those parent styles. Some popular painting forms are Realism, Painterly, Impressionism, Abstract, Photorealism, Abstraction Expressionism, Surrealism, etc. (Vieira, Fabbri, Sbrissa, Da Fontoura Costa & Travieso, 2015). These art styles have distinctive features that differentiate one group from another (Bar, Levy & Wolf, 2014). Every artist follows a different painting style, just as Bangladeshi artists follow different art styles. For example, Shilpacharjo Zainul Abedin reflected his preference for realism ("Portrait of an Artist in Divided South Asia," 2013) while S. M. Sultan practiced Impressionism ("The Impressionist in S.M. Sultan", 2020). As a result, detecting different formations of art style for each artist through human analysis of artwork requires exceptional expertise and has a high risk of errors. In this situation, art historians can use a variety of machine learning and deep learning algorithms to identify the actual artist of a piece of artwork.

Another key application of the artist recognition system is the detection of forgery paintings. Several bogus paintings by well-known artists have been offered as genuine ("Theft, Forgery Rumors Enrage Bangladeshi Art Lovers", 2008). Experts evaluate whether a painting is fake based on personal experience and by studying some paintings and painter's qualities ("Theft, Forgery Rumors Enrage Bangladeshi Art Lovers," 2008). It can decrease the need for an expert by applying deep learning methods to artwork photos, allowing for quick and accurate recognition of original artwork.

In processing an artwork investigation, the extracting features should be similar for artworks by the same artist and distinct in different artists. Expected-signature to identify artists and compare their work to that of other painters. The painting style is an important characteristic that allows specialists to classify paintings (Saleh et al., 2016).

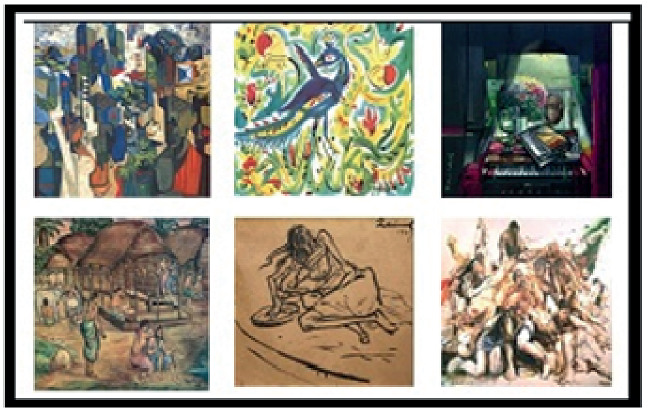


Fig. 1: Artworks of Bangladesh famous artists

Here, in this article, a deep learning approach is used to the problem of image classification depicting fine art paintings of Bangladeshi artists. This classification is intended to detect artists according to their drawing style, color, shape, etc. This is mandatory in making an art catalog, especially when records are becoming digitized. Wikiart is one of the richest datasets, consisting of 150,000 artworks of 2,500 artists (Saleh et al., 2016). Another is Artsy, which has a growing collection and online accessible (Bosch, Zisserman & Munoz, 2008). However, there is no specific dataset containing the artwork of Bengali artists. So, this project worked with a novel dataset created by collecting more than 450 images from Google and various web contents, etc.

This paper aims to use a Convolutional Neural Network to recognize the artist of an artwork. There are two significant contributions to this project:

• Creation of a novel dataset of 450 images of six Bangladeshi painters where collected data is from various resources and then preprocessed to be effective in use.
• Design custom Convolutional Neural Network to investigate the performances to identify artists from artwork images using our dataset.

We divide our paper into five parts. The second section sheds light on some earlier work, while the third piece thoroughly explains the suggested method. The last but second piece, section 4, summarizes the evaluation of the suggested method. Finally, the study finishes in section 5.

2. Related Works:

Artist identification was primarily dependent on human experts, such as volunteers who manually collected and labeled the Wikiart dataset (Saleh et al., 2016), consisting of about 150,000 artworks by 2,500 artists. Experts who manually classify art led the Artsy's Art Genome Project (Bosch et al., n.d.).

CNN has recently achieved substantial advances in several image recognition algorithms (Lombardi, 2005; Saleh et al., 2016). Other research into style classification using CNN has yielded encouraging results. For example, Viswanathan (2017) improves their remarkable performance using a huge dataset encompassing 300 artworks per artist from 57 painters (about 17,000 total paintings) from various styles and times using three different models: baseline CNN, ResNet-18 from scratch, and ResNet-18 with transfer learning. Wang, Lian, Song, Zhang, Zheng, Yue & Ji (2019) developed a Smart Art System (SAS)2 with MobileNet, where the detection module uses a new painting detection algorithm called Single Shot Detection with Painting Landmark Location (SSD-PLL). SSD-PLL can efficiently reduce the influence of complicated background elements on recognition due to MobileNET. There are 7,500 traditional Chinese paintings (TCPs) and 8,800 oil paintings (OPs) in the databases. Another experiment proposes a densely linked neural network architecture with multi-layer feature fusion to identify the painting's author (Bai, Ling, Kai, Qi & Wang, 2019). Unlike classic CNN, Multi-layer Feature Fusion DenseNet (MFDN) uses a combination of shallow and in-depth features to generate conclusions, resulting in more accurate recognition results. Gao, Zhou and Zhang (2020) used two CNNs, VGG19 and ResNet-50, to recognize aesthetic elements in 2300 artwork photos gathered by five artists. Aesthetic characteristics can be subjective or objective. In this paper, specifically, VGG19 prefers to extract subjective features, but ResNet-50 prefers to learn accurate features. Deep CNN is also used to detect artworks in the style and genre (Castellano, 2021). However, some other works have generated features using neural networks followed by SVM classifiers in style identification (Alyannezhadi, Dabbaghan, Moghani & Forghani, 2019; Bar et al., 2014; Sharif Razavian, Azizpour, Sullivan & Carlsson, 2014).

Previous attempts at artist identification were made by extracting hand-crafted features. To verify the works by artistic masters, art historians used a variety of approaches, such as micro-chemical examination of paint materials, canvas thread counting, historical research, etc. (Li, Yao, Hendriks & Wang, 2012). However, art analysts have become intensely interested in analytic machine systems to classify art styles and techniques. That is why, instead of hand-crafting features, we train our CNN model with learned features to get better results.

3. Proposed Methodology:

Here, we developed a ConvNet CNN architecture for recognizing artists. The training and testing phases are the most important parts of our system. The first phase of our model is training, which involves using 80% of our data set to train the model, and the following phase is testing, where 20% of total data is used. There are four steps in the total procedure: 1) Data collection, 2) Preprocessing and data augmentation, 3) CNN Model development, and 4) Feature Extraction and Prediction. Before information extraction and feeding the model, images must be normalized. We attempted to assess two models' classification and prediction accuracy with varying kernel widths. Figure 2 displays our proposed method at a glance:

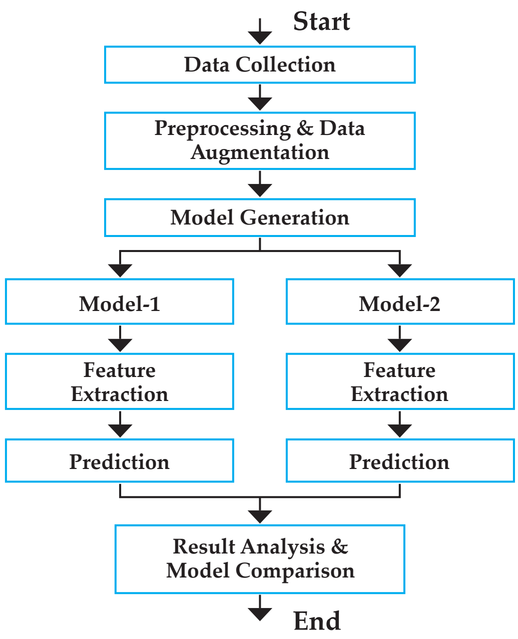


Fig. 2: Proposed Methodology

3.1 CNN Architecture:

We have chosen the Basic Convolutional Neural Network, a deep learning network with widespread use in classification and prediction (Gao et al., 2020). ConvNet is a sequence of layers, and each layer transforms one volume of activations to another through a differentiable function. ConvNet also shows black-box nature in feature generation and mapping, like other deep learning networks. However, CNN's success rate in unique decomposing styles and content components from images forces us to use this algorithm in our artist recognition model.

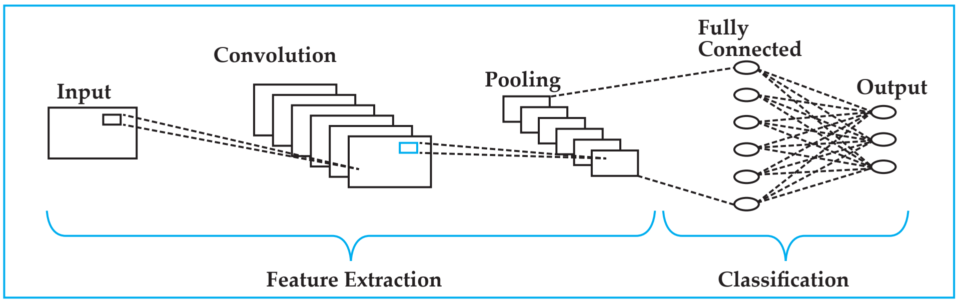


Fig.3: Basic layers of ConvNet CNN

3.1.1 Input layer:

As a multi-dimensional array, the input data is generated from the RGB image. We must normalize the raw pixel value into a range between zero and one. It is critical to improving performance before entering the hidden layer (Gao et al., 2020). The input layer is the first layer in Figure 3.

3.1.2 Hidden Layer :

Convolutional, Pooling, and Fully Connected Layers are hidden layers (Gao et al., 2020). In Figure 3: the second, third, and fourth layers are all hidden layers.

a) Convolution Layer:

By learning characteristics from the image, the convolution layer investigates pixel relationships. This layer generates parameters that are required for feature generation. The input data is delivered in small chunks in this situation. A source image matrix and kernels are inputs for the mathematical operation. The input image is entirely rolled over after being convolved with several filters. A value for stride can govern the movement of filters across the image (Sen, Deb, Dhar & Koshiba, 2021). Stride defines the difference between consecutive kernels. The most typical stride number is 1, but bigger stride values can downsample the feature map. Padding values can be specified to avoid lowering output and losing information. Kernel size, no. of kernels, stride, and padding values work as hyperparameters in CNN design. Tuning of these values affects the progress of accuracy rate. Rectified Linear Unit (ReLU) is used in the CNN layer as an activation function.

$$f(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \quad (1)$$

Equation 1 defines ReLu function. ReLu is a nonlinear function that uses *a max ()* to get an appropriate value rather than the *Sigmoid* or *tanh* function.

b) Pooling Layer:
The pooling layer applies down-sampling to the feature map to reduce its planar dimension. It also introduces distortions and small shifts to translate invariances. Learning parameters are decreased in the output of this layer. In the pooling layer, learnable parameters of pooling operations, if any exist, are not similar as in convolution operations. Max pooling is used in our research. Input feature maps generate patches in max-pooling and collect the maximum patch value while discarding the others. The most typical max-pooling configurations are kernel size of 2 X 2 and a stride value of 2. The feature maps in the planer dimension are downscaled by a factor of two. The depth dimension does not change, but the height and breadth do (Yamashita, Nishio, Do & Togashi, 2018).

c) Fully Connected Layer:
The pooling layer's flattened feature maps are fully connected to other layers with learnable weights. The number of output nodes in a completely connected layer equals the number of classes. The dense layer is another name for this layer. A nonlinear function follows a completely connected layer. In the last layer, the Softmax activation function of equation 2 is used to return the probability distribution of each artist that accumulates to one.

$$\sigma(z)_i = e^{z_j} \sum_{k=1}^{k} e^{z_k} \quad for\ j = 1, \ldots, k \quad \textbf{(2)}$$

The RMSprop optimizer with a 0.00001 learning rate is used in our experiments. The categorical Cross-Entropy loss function is used, which is defined in equation 3.

$$Loss_{cross-entropy}(\hat{y}_i, y) = -\sum_{i=1}^{k} y_i log(\hat{y}_i) \quad \textbf{(3)}$$

Here, $\boldsymbol{K}$ is the total no of classes, $\boldsymbol{y_i}$ is the desired one-hot encoded vector, and $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the model predicted output vector. This loss function tries to maximize the score of the classified correct artists during training.

3.1.3 Output Layer:

The output layer is the third and final layer in Figure 3. The output layer is the last layer of a CNN model, and it generates features incorporating the fully connected layer. Its working principle is the same as a traditional feedforward neural network where logical functions such as softmax are used in production as a classification label (Gao et al., 2020).

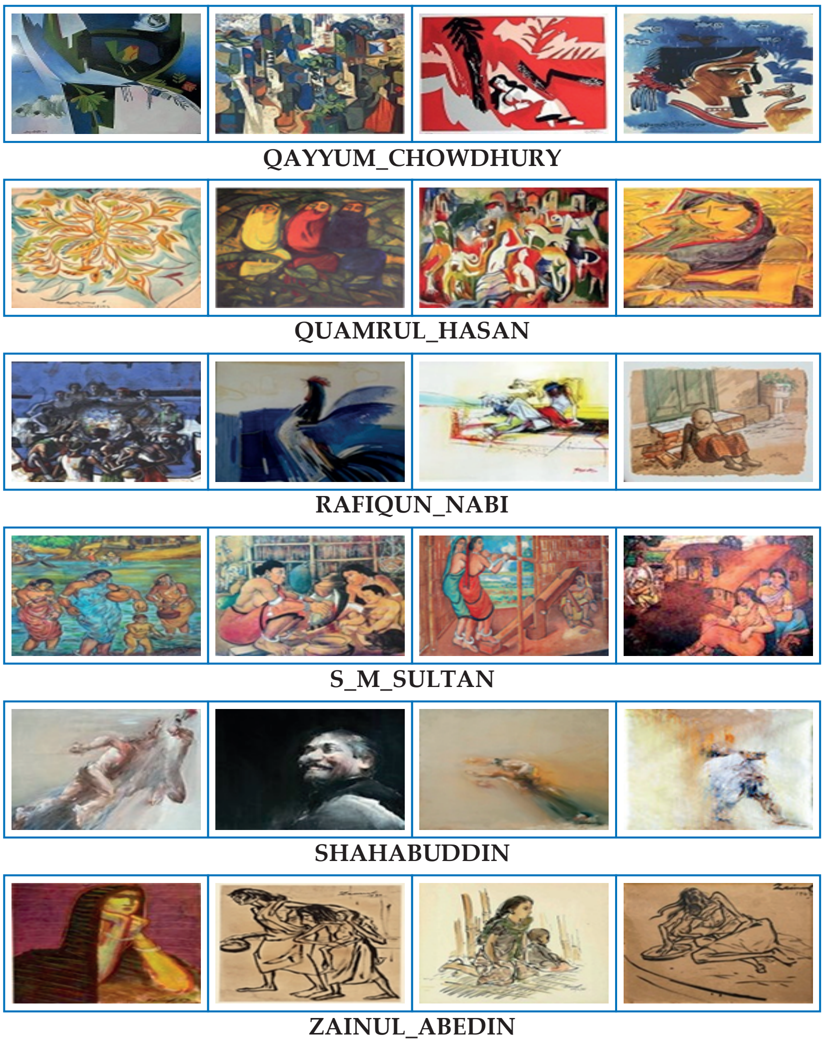


Fig 4. Sample Dataset of ARTWORK of Bengali Artists

3.2 Model Generation:

3.2.1 Data Collection:

We first need to obtain a well-defined dataset of art images in the training phase. However, there is no well-prepared dataset of Bangladeshi artists available. So, we made our data set by collecting raw images from several renowned galleries, Google searching, YouTube videos, and some catalogs of exhibitions.

Our dataset comprises 450 paintings by 6 artists of different periods and styles. Size and shape are not the same for all images. Each image is labeled with the artist's name in a separate .jpeg file. Due to a lack of available data, we tried to collect artworks for an artist with a minimum of 60 or more paintings. Therefore, our dataset includes 6 artists with 450 artworks. Figure 4 represents a sample of our dataset. In our experiment, we divided it into two sets. They are:

a) **Training:** Each artist is assigned a specific level in this phase and creates a hierarchical approach using the generated CNN model. 80% of our dataset was used in this phase.

b) **Testing:** We applied the rest of our dataset to evaluate model performance in this phase.

Artist Name	No. of artwork	Training Set	Testing Set
Qayyum Chowdhury	72	60	12
Quamrul Hasan	71	57	14
Rafiqun Nabi	86	69	17
S. M. Sultan	80	63	17
Shahabuddin Ahmed	78	60	18
ShilpacharjoZainul Abedin	63	50	13

Table 5: Dataset of CNN model

We tried to collect as many paintings per artist to ensure an effective dataset for experiments, but for obvious reasons the training and test set images per artist are unbalanced. Table 5 represents the survey of our Dataset.

3.2.2 Preprocessing and Data Augmentation:

Though we prepared our dataset by collecting images from various sources, it is not large. We modified images before feeding them to the model because the images in the dataset are in multiple sizes and shapes. First, zero centering is applied to the image, followed by normalization. Then, we re-scaled each input image to a size of 150 x 150. And then applied some data augmentation techniques to enhance our dataset, as we know that CNN always works well on large datasets. We applied three augmentation techniques to our image dataset: random flip, shearing with range 0.2, and padding with range 0.2. Small datasets may cause overfitting during training and testing. So, we used data augmentation techniques to our dataset to enhance the data size. We also applied the same methods to ensure stable and reproducible results for the test images. So, the custom dataset helps in avoiding overfitting during testing. Mini-batches (each comprising eight photos) are used to provide input from a reasonable number of images in the dataset. It may take longer to train on a total dataset, but it allows us to use our full dataset without conflict, which improves overall accuracy.

3.2.3 Custom CNN Architecture:

In our work, we designed two models respectively, 1. Model-1 (with a convolution kernel size of 3 X 3), 2. Model-2 (with a convolution kernel size of 5 X 5) to show improvement in class prediction with the same layer configuration except for kernel size. Though those models are simple in design our main motto is to investigate the performance of those simple models on our novel dataset. At starting, The system auto-rescales and processes the first input images with the RGB color channel in model-1, reducing the size to 150 X 150 X 3. The input layer doesn't extract any features from the image. Then the hidden layer starts. Here we use three CNNs in feature generation followed by a flattening and a dense layer to generate the classification output. Filter size, number of kernels, padding, and stride value work as the hyper-parameters in the hidden layer. Tuning of these hyper-parameters concludes with the best value to work with. It can be performed both heuristically and by using some advanced algorithms. But here we heuristically tuned the hyper-parameters to get the best value for our model.

The first convolution block consists of two convolution layers, followed by a Max-pooling. The input image is convolved with some filters and rolled over the whole image in the convolution layer. We tried several different filters in this layer. At 32, we get a satisfactory performance on model 1. As a result of the first convolution, the output shape is 150*150*32. The generated parameter size

is then-
The number of parameters is (3*3*3) +1) *32) = 896, with 1 added as a bias value.

The number of parameters for the second conv2d is (3*3*32) + 1) * 32) = 9248. Here we end the convolution layer of the first CNN. Then it passes to the pooling layer, where we used max pooling. It will process the maximum value in each window. As the pooling layer involves a file of size 2 x 2 with stride 2 on the input image which reduces the input image resolution to half its original size. The convolution block worked with the Rectified Linear Unit (ReLU) operation. The final output generates 75 X 75 X 32 feature maps.

The second convolution block also consists of two convolution layers, but each has 64 kernels, followed by a Max-Pooling. We started with 32 kernels at the first one because the first layer works with low-level features only. So, to get more detailed features, we increased the kernel size and tried out several. Finally, 64 kernels performed satisfactorily best. This block reduces the image size to 37 X 37 pixels at the pulling layer. Because the pooling size has a kernel of 2 X 2 with a stride value of 1 pixel Rectified Linear Unit (ReLU) operation is performed at the end of the block. After pooling, the final output generates 37 X 37 X 64 feature maps.

In the third convolutional block, each convolution layer has 108 kernels. A-Max Pooling operation follows the dense layer and reduces the feature map to 18 X 18 X 108. The number of kernels is chosen heuristically from several trials. We worked on a trial and error basis to get the best performing value of kernels.

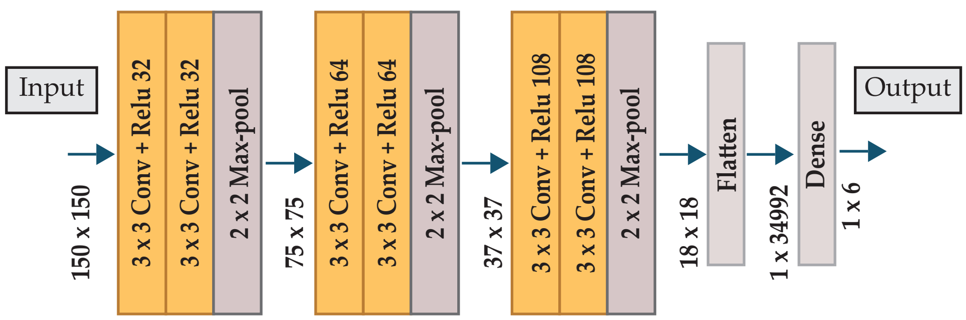


Fig 6. Model-1 architecture

To determine the classification of the input image, the final completely linked layer is connected to the output layer. This has a lot of parameters that can be learned. When compared to the other layers, this layer group generates the most parameters. In classification, the Softmax regression is used; it gives output in normalized form, same as in multi-class classification. It is mostly used to normalize the output of neural networks. It shows the "probability" of the network output's certainty. With a learning rate of 0.0001 and a momentum of 0.9, the RMSProp optimizer is employed.

The final fully connected layer is the dense layer. Image level is condensed into 6 levels in dense layer image level, which is the number of total artists. It gives the output of each CNN model. After successfully generating Model-1, we planned to design Model-2 with the kernel size 5 x 5. Other hyper-parameters such as padding, stride, and multiple kernel values were kept the same as Model-1 to compare the result.

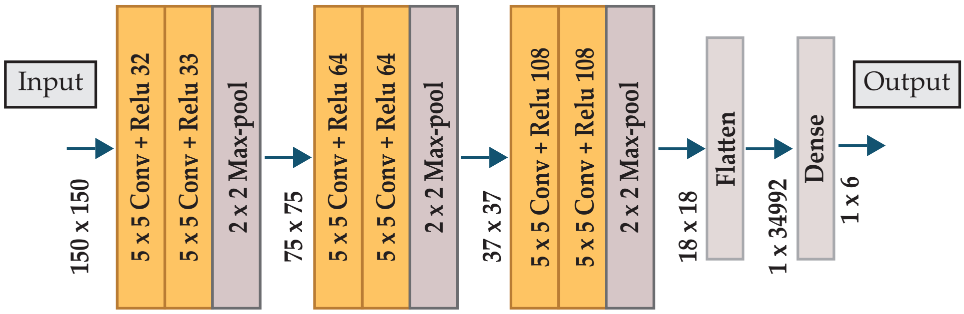


Fig 7. Model-2 Architecture

The design of both models remains the same, but the changed kernel size generates an increased number of parameters in each layer. As in the first conv2d, it will generate: No of Parameters = ((5*5*3) + 1) * 32 = 2432, and pass to the second one. And this time the parameter size is: No of Parameter = ((5*5*32) + 1) * 32 = 25632, which is greater than the parameter size of Model-1's same layer will increase gradually in other layers too. So, this may have a positive effect on accuracy improvement. Model-1 and 2 are depicted in the model architecture depicted in figures 6 and 7.

3.2.4 Feature Extraction & Prediction:

The trained CNN model works as a feature extractor and classifier too. It is effective in extracting abstract features from raw pixels. So, images of datasets work as primary features for the system. Deep neural networks learn features in the fully connected hidden layers. Image transformation occurs in each layer, and new complex features are extracted in every step. The altered image data is passed via the fully connected layers, classified or predicted. Feature visualization is a technique for enhancing activation to make learned characteristics explicit. Following feature extraction, the model predicts its own. There is no use of any classifier such as SVM or others. So, the strengths of the CNN model are that manual feature engineering and individual classifiers are not needed to be used (Yamashita et al., 2018).

4. Result Analysis:

4.1 Hardware Configuration:

Our system configuration is AMD Ryzen 7 2700X Eight-core 3.7GHz Processor, 32 GB RAM, Nvidia GeForce RTX 2060 Super of 8GB GPU Memory. The operating system was Ubuntu version 20.04, and the Backend was managed by Keras using TensorFlow.

4.2 Dataset:

We worked with a self-made novel dataset that consisted of 450 artworks by 6 artists. We tend to focus our projects on Bangladeshi artists, but there are no available public datasets to work with. So, we collected our data from various sources such as Google, art galleries, art exhibitions, YouTube videos, etc. However, the dataset is not big enough to work with CNN, as CNN performs well on large datasets. So we used various data augmentation techniques to enrich the dataset sufficient to work with CNN.

4.3 Model Analysis:

Both models worked on our self-made dataset. The only difference between Model-1 and Model-2 is the kernel size. Other hyper-parameters remain the same in both models, such as padding, stride values, and multiple kernel numbers. However, the kernel size makes a significant difference in parameter generation. The following Tables 8 and 9 clearly show the differences in each layer of parameter generation. Both models, however, use 32, 64, and 108 kernels in three CNNs in a row. We worked on a trial and error basis to choose the number of kernels heuristically to use in each layer. We tried several but were satisfied with the performance of 32 at the first layer. In the first layer, CNN detects the primary parameters of an image such as edge, color, shape, etc. Then in the second layer, CNN, we tried more kernels greater than 32 and found the best at 64. As before, at the third CNN, we used 108 hidden units, which were chosen at random with some other values, and the best performing value was chosen to work with.

Layer (type)	Output Shape	Parameter
Input	(None, 150, 150, 3)	----
conv2d_1 (Conv2D)	(None, 150, 150, 32)	896
conv2d_2 (Conv2D)	(None, 150, 150, 32)	9248
max_pooling2d_1(MaxPooling2)	(None, 75, 75, 32)	0
conv2d_3 (Conv2D)	(None, 75, 75, 64)	18496
conv2d_4 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_2(MaxPooling2)	(None, 37, 37, 64)	0
conv2d_5 (Conv2D)	(None, 37, 37, 108)	62316
conv2d_6 (Conv2D)	(None, 37, 37, 108)	105084
max_pooling2d_3(MaxPooling2)	(None, 18, 18, 108)	0
flatten_1 (Flatten)	(None, 34992)	0
dense_1 (Dense)	(None, 6)	209958

Table 8.: Model architecture of Convolution blocks of Model-1

Layer (type)	Output Shape	Parameter
Input	(None, 150, 150,3)	--
conv2d_1 (Conv2D)	(None, 150, 150, 32)	2432
conv2d_2 (Conv2D)	(None, 150, 150, 32)	25632
max_pooling2d_1(MaxPooling2)	(None, 75, 75, 32)	0
conv2d_3 (Conv2D)	(None, 75, 75, 64)	51264
conv2d_4 (Conv2D)	(None, 75, 75, 64)	102464
max_pooling2d_2(MaxPooling2)	(None, 37, 37, 64)	0
conv2d_5 (Conv2D)	(None, 37, 37, 108)	172908
conv2d_6 (Conv2D)	(None, 37, 37, 108)	291708
max_pooling2d_3(MaxPooling2)	(None, 18, 18, 108)	0
flatten_1 (Flatten)	(None, 34992)	0
dense_1 (Dense)	(None, 6)	209958

Table 9.: Model architecture of Convolution blocks of Model-2

Model-1 generated a total of 442,926 parameters and acquired an acceptable accuracy rate in artist prediction. So to continue the flow of accuracy, we tried our second Model-2 with the kernel size 5 x 5. Model-2 generates 856,366 parameters, almost twice that of Model-1 parameters production. For Model-1 and Model-2, each layer performs the same operation, but the difference is only the filter size. Due to the filteration technique parameter generates, changes occur in the number of parameters. Using a big-sized kernel produces more parameters in each layer, and learning more parameters may positively affect the accuracy rate. Later, the evaluation will show that the accuracy rate of Model-2 outperforms the accuracy rate of Model-1 in prediction.

4.4 Evaluation Metrices:

We introduced classification accuracy (the fraction of paintings whose artists are identified correctly), precision, recall, F1 scores, and support value to evaluate our model. There are four possible cases in classification: True Positive (result is the positive and true case), True Negative (result is the negative and negative case), False Positive (result is the positive but false case), and False Negative (result is the negative but false case). Classification Accuracy is the sum of true positive and true negative cases divided by the total number of samples. So, classification is defined as:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{True\ Positives + True\ Negetives}}{\boldsymbol{True\ Positives + True\ Negetives + False\ Positives + False\ Negetives}} \quad \textbf{(4)}$$

Precision and recall are defined as:

$$\boldsymbol{Precision} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Positives}} \quad \textbf{(5)}$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{True\ Positives}}{\boldsymbol{True\ Positives + False\ Negetives}} \quad \textbf{(6)}$$

The ***F*1** score is a weighted average of precision, and recall values that deals with unbalanced input is defined as:

$$\boldsymbol{F1} = \frac{\boldsymbol{2 * Precision * Recall}}{\boldsymbol{Precision + Recall}} \quad \textbf{(7)}$$

The score is a weighted average of precision, and recall values that deals with unbalanced input are defined as:

4.5 Result Evaluation:

The evaluation value of our CNN Model-1 and Model-2 differs in each parameter. The following two consecutive tables, Tables 10 & 11, show the evaluation matrices for Model-1 and Model-2. The evaluation shows that the average classification accuracy differs for Model 1 and Model-2. The rate with Model-2 is 89%, whereas Model-1 has an 87% accuracy rate.

Class	precision	recall	f1-score	support
QAYUUM_CHOWDHURY	0.73	0.92	0.81	12
QUAMRUL_HASAN	0.71	0.71	0.71	14
RAFIQUN_NABI	1.00	0.82	0.90	17
SHAHABUDDIN	0.95	1.00	0.97	18
S_M_SULTAN	0.94	0.94	0.94	17
ZAINUL_ABEDIN	0.82	0.75	0.78	12
Accuracy			0.87	90
macro avg	0.86	0.86	0.85	90
weighted avg	0.87	0.87	0.87	90

Table 10.: Evaluation report of CNN Model-1

Class	precision	recall	f1-score	support
QAYUUM_CHOWDHURY	0.79	0.92	0.85	12
QUAMRUL_HASAN	0.75	0.86	0.80	14
RAFIQUN_NABI	0.93	0.82	0.87	17
SHAHABUDDIN	0.95	1.00	0.97	18
S_M_SULTAN	0.94	0.94	0.94	17
ZAINUL_ABEDIN	1.00	0.75	0.86	12
Accuracy			0.89	90
macro avg	0.90	0.89	0.88	90
weighted avg	0.79	0.92	0.89	90

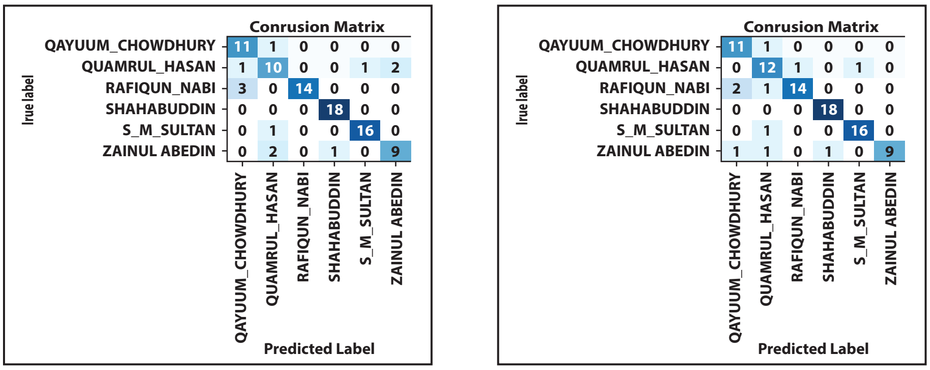
Table 11.: Evaluation report of CNN Model-2

The evaluation report depicts that, Model-2's performance is not highly increased in comparison to Model-1. There differs only 2% in accuracy rate. Though Model-2 generates 856366 parameters, model-1 generates only 442926, which is almost half. So, comparatively, the improvement rate of Model-2 is so low. However, our priority is to obtain more perfection in artist recognition. That is why we take account of Model-2 as an influential one with improved accuracy in artist recognition. There is another point that forces Model-2 to perform better than Model-1. The fact is that the larger kernel size chooses a broader span of an input image in each context while rolling over the whole image. So, the broader region of interest may help in complex feature extraction used in recognition. So, the recognition system performs best with Model-2. Through the evaluation reports, The differences in each case clearly show the improvement in Model-2 over Model-1. Such as, the precision rate is increased in each class in Model-2, except RAFIQUN_NABI, which remains the same as in Model-1. So, similar false-negative cases occur for the RAFIQUN_NABI class. In the recall section, except for the QUAMRUL_HASAN class, each class prediction rate remains the same in Model-2 as in Model-1, whereas it has increased in the QUAMRUL_HASAN class. Due to the change in precision and recall value, F1 of each class is improved in model-2, except for QUAMRUL_HASAN.

Fig.12: (a). Sample dataset for test, (b) True predicted data, (c) false predicted data

In some cases, our models got false predictions in testing. One of the essential factors is the art style. Each artist follows his style incorporating popular art forms. Shilpacharjo Zainul Abedin followed the "Realism" style. His masterpieces "The Struggle," "Rebel Cow," "Famine Paintings," etc., reflect realism. However, after a while, he introduced a new art form named "Bengali style," where the main features were folk forms with their geometric shapes, sometimes semi-abstract representation, and primary color (Doula, 2016). From our sample dataset from figure 3, we get some idea about his painting style. Another critical factor is that one or more artists follow the same style in painting. Shilpacharjo Zainul Abedin and S.M. Sultan both adhered to social realism, so their works may overlap. As a result, these exceptional cases cause some confused predictions.

As we collected our data from Google and some other sites, we could not maintain the quality of the images, which is an excellent factor in image processing. So, some low-resolution image data may cause false predictions in testing. Figure 12 visualizes the sample dataset with accurate and incorrect predictions for both models. Though we tried to enhance our dataset using data augmentation techniques on collected data, we cannot be sure about their positive effect in making our dataset rich. Like any other deep neural network, CNN does not correctly reveal its mid-level working mechanism. The feature extraction and processing are performed in the hidden layer between the input and output layers. So, it is challenging to conclude any specific reason behind a false prediction.

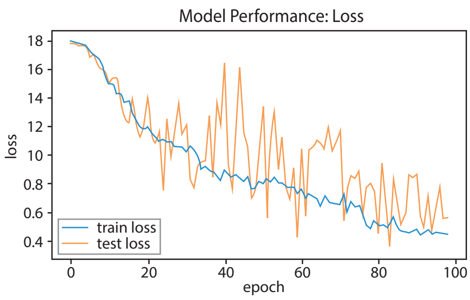


a) *b)*

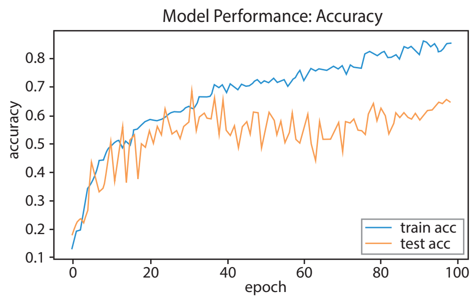
Fig 13. a) and b) showing Confusion matrices for Model-1and Model-2

The confusion matrices for Models 1 and 2 depict the success and failure cases of Models 1 and 2. Each matrix row corresponds to a predicted class, and each matrix column corresponds to an actual class. We tested an equal number of samples from each of the 6 cases in both models.

The same results are shown in the case of QAYUUM_CHOWDHURY, SHAHABUDDIN, and S_M_SULTAN in both models, as kernel change does not affect these three labels. As, 12 artworks by QAYUUM_CHOWDHURY were tested in both models, where only one conflicts with QUAMRUL_HASAN. Of these, 18 by SHAHABUDDIN were used where no false prediction occurred, and 17 artworks by S. M. Sultan were in testing where only one conflicts with QUAMRUL_HASAN. In comparison, Model-2 shows significant improvement, especially in the QUAMRUL_HASAN case, as increased features may have worked as a positive factor. The greater kernel size of Model-2 creates a specific region of interest while rolling over the input image, which is larger than Model-1. It may cause extracting complex features from input. The confusion matrix of Model-1 shows that QUAMRUL_HASAN has some overlapping features with QAYUUM_CHOWDHURY, S_M_SULTAN, and ZAINUL_ABEDIN. However, due to a change of kernel size, the overlapping features were removed for QAYUUM_CHOWDHURY and ZAINUL_ABEDIN classes. A new overlapping occurs with RAFIQUN_NABI, but the overall accuracy improved for QUAMRUL_HASAN class. In ZAINUL_ABEDIN, while true prediction occurs confusion arises as well between different cases. So, after comparing the confusion matrices, it is to be concluded that increased kernel improves the classification only in QUAMRUL_HASAN class. That is why the improvement in accuracy rate is not much significant.



a) Loss graph



b) Accuracy graph

Fig.14: Loss and accuracy graph for Best model (for 100 epochs)

Graphs of loss and accuracy for our best model Model-2, as shown in Figures 14(a), (b), demonstrate that test accuracy equaled training accuracy for some initial epochs. After some epochs, training accuracy was higher than test accuracy. The loss graph shows that the training loss and the test loss are decreased in a downright quick way till 20 epochs, after which the decrement goes slower than before. The increasing accuracy rate for training and testing is almost the same until 20 epochs. Both graphs are generated for 100 epochs as the model shows its training and testing nature and the best performance in this range.

4.6 Comparison With Existing Works.:

Although our models are not new in work, our dataset is unique. So, to test the effectiveness of our designed model and the dataset, we tried a recent work to compare with. We compared our model's performance with Viswanathan (2017) as they used three models in artwork recognition: Baseline CNN, ResNet-18 from Scratch, and ResNet-18 with transfer learning. Though ResNet is a present state-of-the-art model, it works with Transfer learning to fine-tune their accuracy. Our design only tried to investigate the improvement nature of custom CNN structure on our novel dataset. Since we didn't introduce transfer learning in our designed models yet, we applied their custom Baseline CNN to our dataset. The result comparison shows in the following table-

Experiment	Accuracy rate (%)
Baseline CNN (Viswanathan, 2017)	84%
ConvNet Model-1	87%
ConvNet Model-2	89%

Table 15: Comparison of results in the field of artist recognition of paintings Experiment

Viswanathan (2017) selected a large dataset containing 300 paintings per artist from 57 artists (about 17,000 total paintings) from various styles and periods. We tried to test our trained model using his dataset, but unfortunately, it is not public. So among the publicly existing datasets, we collected a small part of the famous **Wikiart** Dataset ("Painter by Numbers," 2016), which consists of 6 artists with the most artworks. Due to hardware availability, we could not use the whole dataset, as it is a rich one with a total of 120,000 paintings of 2,300 artists of different periods and styles. We applied our best-trained model with this dataset, and the result is shown in the following table-

Experiment	Accuracy rate (%)
ConvNet Model-2 (Wikikart dataset) ("Painter by Numbers", 2016)	68%
ConvNet Model-2 (own dataset)	89%

Table 16: Comparison of results in the field of artist recognition of paintings Experiment

The accuracy rate of our best model on this dataset is only 68%, which is comparatively lower than our experiment result on our novel dataset. The result may be decreased due to the generalization problem.

4.7 Discussion:

In this work, we used simple models of CNN to get the classification and recognition systems for Bangladeshi artists. However, the lack of available data made our job difficult as we know that deep learning networks work well on large datasets. We will try to overcome this issue in the future. Another critical issue is the hyperparameter tuning in the CNN layers. We used heuristic hyper- parameter tuning and chose random parameters on a trial and error basis. We plan to work with advanced algorithms to get the best hyperparameters in our model. Moreover, we will use state-of-the-art models to our dataset and introduce transfer learning with our models to fine-tune the result as future work because fine-tuning a pre-trained CNN yields the best result and outperforms the state-of-the-art results (Ren Tan, Seng Chan, Aguirre & Tanaka, 2016). We will try to add some interpretable machine learning algorithms, saliency graphs, etc., to visualize the feature mapping procedure of our models.

5. Conclusion:

This paper developed two convolution neural network architectures for an artist recognition system using paintings. CNN is a deep learning network that is supervised and universal. It has a high accuracy rate and is highly training- efficient (Guo, Zhang, Yin, Hu, Zou, Xue & Wang, 2020). That is why we chose the CNN model in developing an artist recognition system. In our system, the system hierarchy is developed before the classification accuracy rate of the trainee class. So, the primary input feature clearly expresses the distinction between classes and associated levels. Our proposed system is evaluated through a self-made dataset of 450 artworks by six famous Bangladeshi painters. The accuracy rate of our artist recognition system is 87% for Model-1 and 89% for Model-2. We will try to quantify the prediction rate using image style as a feature versus the image itself in future work. We will try to include some updated models of CNN in the design and use some advanced models like Grid Search in hyper-parameter tuning. We want to enrich our dataset by including artists with a moderate collection of paintings and classifying them with higher accuracy than before.

References:

Abry, P., Klein, A. G., Sethares, W. A., & Johnson, C. R. (2015). Signal Processing for Art Investigation. *IEEE Signal Processing Magazine*, *32*(4), 14–16. Retrieved from https: //doi.org/10.1109/MSP.2015.2419311

Alyannezhadi, M. M., Dabbaghan, H., Moghani, S., & Forghani, M. (2019). A Painting Artist Recognition System Based on Image Processing and Hierarchical SVM. 2019 *IEEE 5th Conference on Knowledge-Based Engineering and Innovation, KBEI 2019*, 537–541. Retrieved from https://doi.org/10.1109/KBEI.2019.8734911

Bai, R., Ling, H., Kai, Z., Qi, D., & Wang, Q. (2019). Author recognition of fine-art paintings. *Chinese Control Conference, CCC, 2019-July*, 8513–8518. Retrieved from https://doi.org/10.23919/ChiCC.2019.8865492

Bar, Y., Levy, N., & Wolf, L. (2014). Classification of Artistic Styles Using Binarized Features Derived from a Deep Neural Network. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8925*, 71–84. Retrieved from https://doi.org/10.1007/978-3-319-16178-5_5

Bosch, A., Zisserman A., and Munoz X. (2008). Image classification using ROIs and multiple kernel learning. *International Journal of Computer Vision*, Retrieved from http://eia.udg.es/~aboschr/publicacions/bosch08a_preliminary.pdf

Castellano, G. (2021). Deep learning approaches to pattern extraction and recognition in paintings and drawings: an overview. *Neural Computing and Applications*, *33*(19), 12263–12282. Retrieved from https://doi.org/10.1007/s00521-021-05893-z

Doula, S. (2016). *The Legend Artist Zainul Abedin*(Research Report). Retrieved from https://www.academia.edu/33007818/The_Legend_Artist_Zainul_Abedin_by_Sultana_Doula

Gao, J., Zhou, H., & Zhang, Y. (2020). The Performance of Two CNN Methods in Artworks Aesthetic Feature Recognition. *ACM International Conference Proceeding Series, 28, 289–296*. Retrieved from https://doi.org/10.1145/3383972.3383974

Guo, Y., Zhang, J., Yin, C., Hu, X., Zou, Y., Xue, Z., & Wang, W. (2020). Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming. *Discrete Dynamics in Nature and Society*. Retrieved from https://doi.org/10.1155/2020/2479172

Johnson, C. R., Hendriks, E., Berezhnoy, I. J., Brevdo, E., Hughes, S. M., Daubechies, I., Li, J., Postma, E., & Wang, J. Z. (2008). Image processing for artist identification: Computerized analysis of Vincent van Gogh's painting brushstrokes. *IEEE Signal Processing Magazine, 25*(4), 37–48. Retrieved from https://doi.org/10.1109/MSP. 2008. 923513

Li, J., Yao, L., Hendriks, E., & Wang, J. Z. (2012). Rhythmic brushstrokes distinguish van Gogh from his contemporaries: Findings via automated brushstroke extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(6), 1159–1176. Retrieved from https://doi.org/10.1109/TPAMI.2011.203

Lombardi, T. E. (2005). *The Classification of Style in Fine-Art Painting* (Dissertation). Retrieved from https://digitalcommons.pace.edu/dissertations/AAI3189084

Painter by Numbers (2016, April 30) *Kaggle.com*. Retrieved February 25, 2022, from https://www.kaggle.com/c/painter-by-numbers

Portrait of an artist in divided South Asia(2013, January 6). *DAWN.COM*. Retrieved February 26, 2022, from https://www.dawn.com/news/776951/portrait-of-an-artist-in-divided-south-asia

Ren T. W., Seng C. C., Aguirre, H. E., & Tanaka, K. (2016). *Ceci n'est pas une pipe: A Deep Convolutional Network for Fine-art Paintings Classification* (Research Report). Retrieved from http://chttp://cs-chan.com/doc/ICIP2016.pdf

Saleh, B., Abe, K., Arora, R. S., & Elgammal, A. (2016). Toward automated discovery of artistic influence. *Multimedia Tools and Applications, 75*(7), 3565–3591. Retrieved from https://doi.org/10.1007/s11042-014-2193-x

Sen, A., Deb, K., Dhar, P. K., & Koshiba, T. (2021). Cricshotclassify: An approach to classifying batting shots from cricket videos using a convolutional neural network and gated recurrent unit. *Sensors, 21*(8). Retrieved from https://doi.org/10.3390/s21082846

Sharif R. A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition*, 2014 IEEE conference on computer vision and pattern recognition workshops (pp. 806–813) Retrieved from https: //doi.org/1403. 10.1109/CVPRW.2014.131

The Impressionist in S.M. Sultan (2020, May 20). *Dhaka Courier* Retrieved February 26, 2022, from https://dhakacourier.com.bd/news/Culture/The-Impressionist-in-SM-Sultan /2436

Theft, forgery rumors enrage Bangladeshi art lovers (2008, February 27) *Reuters*. Retrieved November 8, 2021, from https://www.reuters.com/article/us-bangladesh-antiques-idUSSYD7101720080228

Vieira, V., Fabbri, R., Sbrissa, D., Da Fontoura Costa, L., & Travieso, G. (2015). A quantitative approach to painting styles. *Physica A: Statistical Mechanics and Its Applications, 417*(November 2013), 110–129. Retrieved from https://doi.org/10.1016/j.physa.2014.09.038

Viswanathan, N. (2017). *Artist Identification with Convolutional Neural Networks* (Dessertation). Retrieved from http://cs231n.stanford.edu/reports/2017/pdfs/406.pdf

Wang, Z., Lian, J., Song, C., Zhang, Z., Zheng, W., Yue, S., & Ji, S. (2019). SAS: Painting detection and recognition via smart art system with mobile devices. *IEEE Access, 7*, 135563–135572. Retrieved from https://doi.org/10.1109/ACCESS.2019.2941239

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. In *Insights into Imaging*, *9*(4), 611–629. Retrieved from https://doi.org/10.1007/s13244-018-0639-9