Journal of High School Science

Artwork recognition based on ResNet models with an added batch normalization layer

Shan A

Submitted: August 8, 2024, Revised: version 1, October 20, 2024
Accepted: October 28, 2024

Abstract
Artwork recognition is an important and unique application of image recognition networks. Advanced art recognition networks can not only help to construct digital archives for artwork, but also serve as educational tools for people to use to learn about art. Compared to other types of image recognition tasks, artwork recognition has unique challenges. Not only can different artists have similar artistic styles, but artwork from the same artist can also differ significantly, making it difficult to categorize artwork. Convolutional Neural Networks have always been used to perform image recognition tasks. This research proposes two ResNet50 and ResNet152 models with an added batch normalization layer. Many algorithms that mitigate overfitting during training processes are also explored in this paper. The ResNet152 model outperformed the ResNet50 model, reaching a validation set accuracy of 88% and a test set accuracy of 86%. Compared to different artwork recognition models in previous researches, the network introduced in this paper not only has a simpler architecture, but also generalizes better on a larger comprehensive dataset.

Keywords
Computer Science, Machine Learning, Convolutional Neural Networks, Artwork recognition, Image recognition, ResNet, Overfitting, Batch normalization, Paintings, Digital archives

Allen Shan, Tsinghua International High School. Zhong Guan Cun North Street, Beijing, China 100084. allenshan.yitian@gmail.com

1 Introduction

Machine learning has significantly advanced the field of image recognition, enabling the development of complex networks capable of extracting nuanced patterns from data. Among the various applications of image recognition, the identification and classification of artwork is particularly challenging, yet consequential. A major challenge in artwork recognition is that artworks from different artists can have similarities and artworks from the same author can also have different styles. Developing models that can accurately recognize and categorize artwork greatly impacts cultural heritage preservation, art market dynamics, and art education. This research paper focuses on the development and training of artwork recognition models based on ResNet50 and ResNet152 architectures, which are both deep convolutional neural networks. This paper will also explore various ways to overcome overfitting, a common challenge faced during machine learning, to enhance the model's generalization ability.

Artwork recognition is important for several reasons, 1] Art Archives Development: With the growing trend of digitizing art collections, advanced artwork recognition systems enable more efficient organization and retrieval of artworks in digital archives. Archives enhance the accessibility of artwork for researchers, historians, and the general public, promoting a broader appreciation and understanding of art. Digital archives also ensure that important cultural heritage is preserved for future generations by cataloging and safeguarding valuable artworks, preventing loss or theft, and facilitating recovery efforts. 2] Art Market and Authentication: In the art market, the ability to authenticate the provenance of artworks is crucial. Accurate recognition models assist in distinguishing original works from forgeries, thus maintaining the integrity of the market and protecting the interests of collectors and institutions. 3] Educational Tools: Often, people come across artworks they do not know in galleries or museums. Artwork recognition apps are able to provide information about the artwork, and thus help them learn more about the artwork, the artist, and art history.

This research aims to develop two artwork recognition models using ResNet50 and ResNet152, taking advantage of their deep architecture to capture complex patterns from diverse artworks. ResNet models are chosen for this research because they are most commonly used for art recognition tasks, and have an excellent reputation in image recognition tasks. The primary objectives of this research were 1] Model Development and Training: Train ResNet50 and ResNet152 models on a comprehensive dataset of artworks. This involved preprocessing the data, constructing the model, applying different algorithms, and performing training tasks to achieve higher generalization ability. 2] Overfitting Mitigation: Overfitting is a common difficulty faced in machine learning, where a model performs well on the training data but fails to generalize to the validation

and testing sets. This research explored various strategies to mitigate overfitting such as 1] Data Augmentation: Techniques such as color jittering, resizing, and rotation can artificially expand the training dataset and introduce variability, hence improve the model's generalization ability. 2] Optimizers and Schedulers: Algorithms like Adaptive Moment estimation (Adam), Root Mean Square Propagation (RMSprop), Stochastic Gradient Descent (SGD), and ExponentialLR help adjust the model's parameters and learning rate to minimize the loss function and improve model performance during training tasks. 3] Batch Normalization: An algorithm that improves training stability and accelerates convergence through normalizing the activations of each layer in the network while preserving its meaningful information and patterns.

The research outlined in this paper sought to find the highest potential of the performance of ResNet50 and ResNet152 based artwork recognition models after applying different machine learning algorithms. By addressing the challenges of overfitting and enhancing model stability, this research aimed to contribute to both the academic and practical usages of art recognition.

2 Related work

Work using the same Kaggle dataset utilized in this research

Protik et al. used images in this dataset that belonged to three artistic style categories: Expressionism, Impressionism, and Surrealism (1). They utilized transfer learning on the ResNet50 model to enhance its performance. When using transfer learning, a model is first trained on a large dataset to learn general features that are common across datasets used in the first and second tasks. This pre-trained model is then fine-tuned with a smaller, task-specific dataset, which allows the model to adjust its weights and parameters to better suit the new task. The advantages of using transfer learning are significantly reduced training time and improved model performance on the second fine-tuned model. With this method, they achieved a 71% accuracy in categorizing artworks into the three artistic styles.

Gao et al. used five classes in this dataset that had the most number of images (2). Many algorithms like kernel visualization, grad-cam heat-map, confusion matrix, and style transfer were implemented to explore how the ResNet50 model extracted features from images and its performance on the dataset. In their hands, the Resnet50 model resulted in an 82.31% accuracy on the validation set.

Work using other datasets on ResNet models

Menis-Mastromichalakis et al. innovated two training methods for a total of 8 CNN models: including VGG16, VGG19, ResNet50, ResNet152, Inception V3, DenseNet121, DenseNet201, and Inception ResNet-V2 (3). Two datasets were used in this research, the Paintings Dataset for Recognizing the Art Movement (Pandora 18K) collection consisting of 18038 artworks divided into 18 classes and the WikiArt collection consisting of 80039 artworks

divided into 27 classes. The first method in this research was a Simple Ensemble, which involved creating a data generator that fed multiple versions of the input image to each individual network. Then, the average probability of the image belonging to each class was calculated from the outputs of the eight individual networks, forming the final output of the model. The second method introduced was Stacking Ensemble, where the same data generator was used to feed the networks. Subsequently, the outputs of each model were merged into a vector of size N×M, where N is the number of models and M is the number of classes. This vector was then fed to a meta-classifier trained to accurately combine the merged outputs and determine the class to which initial input image belonged. With these two methods that combined the results of many CNN models to achieve a more comprehensive training result, their research achieved a highest test set accuracy of 72.47% with the Stacking Ensemble method on the Pandora 18K dataset.

Del Chiaro et al. used features extracted from CNN models (ResNet50, ResNet101, ResNet152, VGG16, and VGG19) pre-trained on ImageNet to feed into a shallow classifier (4). This research used a NoisyArt dataset consisting of 89095 artworks divided into 3120 classes. Multiple noise mitigation techniques were tested on the models, including labelflip absorption, entropy scaling, and gradual bootstrapping, to improve classification accuracy. They achieved a highest validation accuracy of 68.71% and a test accuracy of 77.09% after removing outlier classes which exhibited uncharacteristically high entropy.

3 Materials and Methods

Dataset

The dataset utilized in this research was the entire "Best Artworks of All Time" dataset on Kaggle, which consists of 8,446 artworks by 50 famous artists from Europe and Northern America. Each class is named after a unique artist and contains all the artworks by that artist [5]. The artworks in this dataset come from a wide variety of artistic styles and periods, ensuring diversity in the visual features presented to the network, as well as testing the model's ability to effectively distinguish between different artistic styles. The rich and diverse images of this dataset provide a solid foundation for developing models capable of nuanced understanding and analysis of varied artworks.

Data augmentation and preparation

Each of the 50 classes in the dataset was split into training, validation, and testing sets in a 70/10/20 ratio. This stratified split minimized the bias between classes, such as what would occur when compared to splitting the dataset as a whole.

Artworks have distinctive characteristics that set them apart from other types of data in the context of image recognition model training. These include significant variability in style, texture, color, and the presence of abstract and subjective elements that challenge conventional pattern recognition algorithms.

Unlike typical images, which have more consistent and predictable features, artworks often exhibit a broad spectrum of artistic expressions, from realistic to abstract. This complexity necessitates specialized data augmentation methods to preserve the integrity of artistic elements. Standard data augmentation techniques may disrupt the color palette, proportions, completeness, and orientation of the artwork, leading to the model misinterpreting important characteristics of the artwork. Implementing unsuitable data augmentation methods can lead to the model overfitting during longer training tasks (> 80 epochs) and lower training and validation accuracy during shorter training tasks (< 50 epochs).

Due to these characteristics of artwork, some common data augmentation methods cannot be used for artwork recognition. The following data augmentation methods are unsuitable for categorizing artworks, 1] Excessive Color Jitter (parameters brightness, contrast, saturation, and hue are > 0.5) and Invert - Color is one of the most important characteristics of artwork. Using excessive color jitter or inverting results in the model not being able to correctly learn the distribution of color and brightness in the artwork. 2] Shearing, Erasing, and Excessive Cropping - Applying these data augmentation methods have the potential to remove important parts of the artwork, as well as changing its proportion and structure, causing the model not being able to correctly learn all the characteristics of the artwork. 3] Gaussian Blur and Excessive Noise - These data augmentation methods decrease the level of detail in the artwork, causing sparsity of detail in the processed artworks. 4] Excessive Rotation and Vertical Flip - These data augmentation methods impact the learning of artwork characteristics under certain circumstances. If the art piece is symmetrical or can only be viewed in one direction, then rotating or flipping the piece could cause its structure to completely change.

Comparative analysis of the training results of implementing suitable and unsuitable data augmentation methods

Figures 1 and 2 show two sets of loss-epoch and accuracy-epoch graphs and the large difference in loss and accuracy between the two sets. Figure 1 shows the training results of a Resnet50 model using unsuitable data augmentation methods, including over cropping, rotation, erasing, and vertical flipping, along with appropriate methods like horizontal flipping, slight color jitter, and random affine, causing high loss and low accuracy. On the other hand, Figure 2 shows the training results of a Resnet50 model only using data augmentation methods suitable for artwork. The difference in accuracy between these two models reached 30% in a 40 epoch training task.

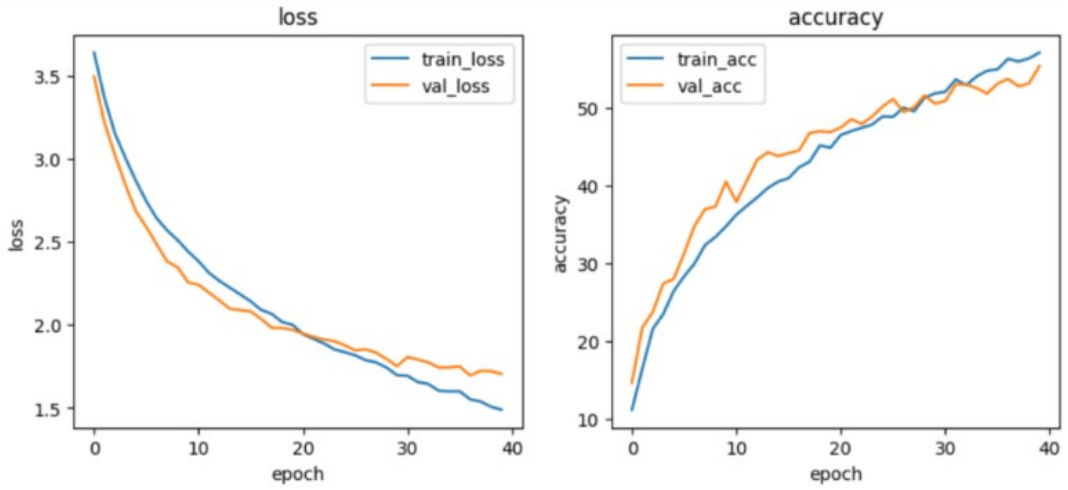

Figure 1: Training results of using unsuitable data augmentation methods

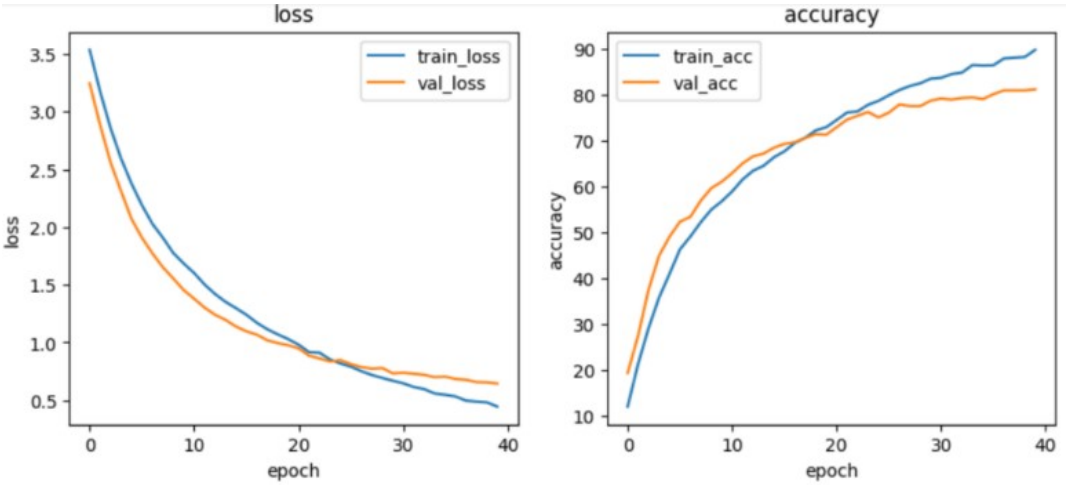

Figure 2: Training results of using suitable data augmentation methods

One important data augmentation method used to increase model accuracy is resizing all images to a uniform size. The model shown in Figure 1 was trained using 'transforms.Resize (224)', which resizes the shorter side of the image to 224 pixels while keeping the original aspect ratio of the image. However, the model in Figure 2 is trained using 'transforms.Resize (224, 224)', which resizes all images to 224*224 pixels, changing the original length-to-width ratio of the image and contributing to the large increase in validation accuracy. Even though resizing images can affect the structure of some artworks (for example the proportions of the figure's head in Cabeza Azteca shown in Figure 3), keeping all images the same size during training is important since it increases the efficiency, stability and accuracy of the model.

Figure 3: Cabeza Azteca by Roberto Montenegro

Figure 4: The features importance heat-map of Cabeza Azteca by Roberto Montenegro

Comparing Figures 1 and 2 shows the importance of keeping all the images in the dataset the same size and choosing the appropriate data augmentation methods for a specific dataset. Changing data augmentation methods can be a pertinent way to overcome overfitting in a model when other methods do not have a significant effect. Depending on the characteristics of the dataset, applying excessive changes to the images may or may not achieve the goal of increasing the model's generalization ability.

Analysis of features importance heat map

Figures 3 and 4 are an image of Roberto Montenegro's Cabeza Azteca and its features importance heat map. The areas of highest significance are centered around the intricate patterns and contours of the figure's head and facial features. These regions are emphasized due to their unique structural elements and contrasting colors, which are critical for distinguishing this artwork from others.

To enhance the model's performance, data

transformation methods such as horizontal flipping are used, allowing the model to recognize the artwork's features regardless of orientation. Similarly, slight color jittering helps the model adapt to minor variations in color intensity and lighting. Random affine introduces subtle changes in perspective, helping the model generalize better across different viewing angles.

These methods not only assist the model in identifying the most important features of the artwork but also prevent the loss of critical information—such as the figure's head in Cabeza Azteca—and avoid distortion of spatial structures, such as can arise from cropping, rotation, and blurring.

Model architecture

There were two models used in this research, ResNet50 and ResNet152, which are both deep convolutional neural networks. The structure of ResNet is based on the concept of residual learning, where shortcuts are added between layers to minimize vanishing and exploding gradients (7). As seen in Figure 5, ResNet models consist of an initial convolutional layer followed by four residual blocks containing more convolutional layers, batch normalization layers, and ReLU activation functions. Each residual block consists of three convolutions: a 1x1 convolution that reduces the number of channels, a 3x3 convolution, and another 1x1 convolution that recovers the original number of channels. The final layer of the model that extracts the learned features and outputs the class that the image is most likely to belong to includes an average pooling layer, a fully connected layer, and a softmax activation function.

ResNet models extract features from images through a series of stages. First, the input image is resized to a fixed size of 224x224 pixels. It then passes through an initial convolutional layer which filters low-level features like edges and textures. The core of ResNet consists of multiple residual blocks, each containing convolutional layers that learn feature representations, batch normalization layers that stabilize training, ReLU activation functions for non-linearity, and skip connections that add the input of each block to its output. As images progress through these blocks, ResNet captures increasingly complex and abstract features, moving from simple shapes to high-level abstract patterns through a process called feature hierarchy. Pooling layers are applied after the residual blocks to downsample the feature maps, reducing dimensionality while preserving essential information. Finally, the features are flattened and passed through a fully connected layer to become a feature vector, which outputs the final prediction.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5: Architecture of ResNet Models (5)

ResNet152 is expanded from ResNet50, with the largest difference being the number of layers. ResNet152 has 152 layers, while ResNet50 has 50 layers. This increased depth allows ResNet152 to extract more complex features from images, leading to higher performance on more complicated image recognition tasks. However, ResNet152 requires a better training environment and a longer training time than ResNet50. Despite these disadvantages, the deeper architecture of ResNet152 still makes it the preferred choice for training tasks where greater accuracy is desired and sufficient computational resources are available.

Training process

The models in this work were trained on Goggle Colab with a T4 GPU. The models were trained with an initial learning rate of $1x10^{-3}$, num_workers of 4, batch size of 16, and a cross-entropy loss criterion. The goal in machine learning tasks is to minimize the loss, which means trying to get the prediction as close as possible to the actual class of the input.

4 Results

Different optimizers

Optimizers adjust the model's parameters to minimize the loss function during training tasks. Optimizers update the parameters based on gradients computed by backpropagation. Different optimizers have different strategies for updating parameters, which affect the training process and results differently. For example, SGD updates parameters with a fixed learning rate and converges slowly, while Adam uses an adaptive learning rate, resulting in faster convergence. Choosing different optimizers can lead to differences in the model's training speed, convergence stability, and overall accuracy.

Adam and AdamW optimizer

Adam and AdamW were the first two optimizers used in this work. Adam implements adaptive learning rates by estimating two moments of the gradients (8).

The first moment is the mean of the gradients and the second moment is the variance of the gradients. AdamW is a variant of Adam that separates the weight decay step from the parameter update (9). AdamW directly applies weight decay to the weights after the parameter update, resulting in improved model performance in most cases. However, AdamW did not outperform Adam on the two models in this work. This may be due to the model already having sufficient regularization when using the Adam optimizer and AdamW's higher sensitivity to the learning rate.

Equationset 1 shows how the Adam optimizer updates model parameters. M_t is the first moment; V_t is the second moment; μ is the initial learning rate; β_1 and β_2 are hyper-parameters that control the decay rate of the mean and variance of the gradients; w_t is the parameter; ∇w_t is the gradient loss of the parameter; ϵ is a small constant; and t is the time step.

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1) * \nabla w_t$$
$$v_t = \beta_2 * v_{t-1} + (1-\beta_2) * (\nabla w_t)^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^{\,t}}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^{\,t}}$$
$$w_{t+1} = w_t - \frac{\eta * \nabla w_t}{\sqrt{\hat{v}_t + \varepsilon}}$$

Equationset 1: Equations the Adam Optimizer uses to update parameters

After the first and second moments are calculated, the biased-corrected moments are calculated. Lastly, the parameter is updated. For the AdamW optimizer, weight decay is applied to w_t before the final update of the parameter through Equation 2 below where λ is a predetermined weight decay constant.

$$w_t = w_t - \mu * \lambda * w_t$$

Equation 2: Equation the AdamW Optimizer uses to perform weight decay

RMSprop optimizer

The third optimizer experimented with in this work was RMSProp. RMSProp also adjusts learning rates adaptively (10). This helps normalize the updates and ensures that the learning rate is appropriate for each parameter in the model.

Equationset 3 shows how the RMSprop optimizer updates parameters. η is the initial learning rate; r_t is the mean square gradient; β is a hyper-parameter that controls the decay rate of the mean square gradient; w_t is the parameter; g_t is the gradient loss of the parameter; ϵ is a small constant; and t is the time step.

$$g_t = \frac{\partial C(t,o)}{\partial w_t}$$

$$r_t = \beta * r_{t-1} + (1-\beta) g_t^2$$

$$\Delta w_t = \frac{\eta * g_t}{\sqrt{r_t + \varepsilon}}$$

$$w_{t+1} = w_t - \Delta w_t$$

Equationset 3: Equations the RMSProp Optimizer uses to update parameters

After the mean square gradient is calculated with the gradient loss, the update amount (Δw_t) is calculated. Lastly, the parameter is updated.

SGD optimizer

The last optimizer experimented in this work was SGD. It was found to have the best performance among all the optimizers. SGD updates the model's parameters by computing the gradient of the loss function and then moving the parameters in the opposite direction of the gradient (11). This update is done in mini-batches of data, which helps the model escape local minima and find a global minimum. While SGD converges slower than Adam and RMSprop, it tends to have better generalization ability. In this work, models using Adam, AdamW, and RMSProp all showed significant overfitting in the first 15 epochs of training, while the model using SGD did not show signs of overfitting until the 50th epoch of the training. For artwork recognition, where the dataset is nuanced due to the colors and styles of the artwork, SGD's simplicity leads to better model performance. The inherent noise in SGD updates also contributes to regularization, preventing overfitting in the training dataset and helping the model generalize well to the validation and test sets. Equation 4 shows how the SDG optimizer updates parameters in an one step formula. η is the initial learning rate; ω_t is the parameter; g_t is the gradient loss of the parameter; and t is the time step.

$$\omega_{t+1} = \omega_t - \eta * g_t$$

Equation 4: Equation the SGD Optimizer uses to update parameters

Schedulers

Schedulers adjust the learning rate during training tasks according to a predefined schedule (12). The learning rate controls the size of the steps the optimizer takes to reach the minimum of the loss function. Corrections to the learning rate can largely impact the model's convergence speed and performance. Common schedulers include StepLR, which decreases the learning rate by a factor every few epochs (e.g. multiplying 0.1 to the learning rate every 20 epochs) (13); MultiStepLR, which allows specifying the range of epochs in which to apply StepLR (e.g. multiplying 0.1 to the learning rate every 20 epochs for epochs 20 to 80); ExponentialLR, which decreases the learning rate exponentially (e.g. multiplying 0.99 to the learning rate after every epoch); and ReduceLR OnPlateau, which decreases the learning rate when a metric has stopped improving (e.g. multiplying 0.1 to the learning rate if validation loss does not decrease by 0.05 in 10 epochs).

The scheduler eventually used in this work was ExponentialLR, which smoothly decreases the learning rate. This smooth decrease helped the model to converge stably. The key to fully simulating the potential of the ResNet152 model was the scheduler. ResNet152 is very sensitive to the learning rate, since it is more likely to enter a local minima than ResNet50. Without a scheduler, the ResNet152 model's accuracy was outperformed by the ResNet50 model in this work.

Batch normalization

To increase the performance of the model used in this work, an extra batch normalization layer was added to the two ResNet models. Batch normalization improves training stability and accelerates convergence by normalizing the activations at each layer of the model (14). After batch normalization, the mini-batch of activations will have a mean of 0 and variance of 1, as calculated using Equationset 5.

$$\mu = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} \mathrm{x}_i$$

$$\sigma^2 = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} (x_i - \mu)^2$$

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

Equationset 5: Equations the Batch Normalization algorithm uses to normalize activations

In this equationset, μ is the mean, σ is the variance, X is the mini-batch of activations to be input into a particular layer, m is the length of the mini-batch, and ε is a small constant. The batch normalization algorithm has two other parameters, scale (γ) and shift (β), which allow the activation to keep carrying meaningful and complex patterns after being normalized through Equation 6.

$$\hat{y}_i = \gamma \hat{x}_i + \beta$$

Equation 6: Equation the Batch Normalization algorithm uses to finalize activations

After implementing batch normalization, the validation and test accuracy of the models improved by 5%, demonstrating the ability of batch normalization to process overfitting problems. ResNet models contain batch normalization layers. To add another batch normalization layer, a class needed to be defined to customize a pre-trained ResNet50 or ResNet152 model. The code to define this class is shown in Figure 6. Note that the class here refers to the Python class, not to be confused with the classes in the dataset, which represent the categories fed into the model.

Final results of trained model

The final results of the two models trained in this research were evaluated by various metrics, such as validation and test set accuracy, precision, recall, F1, AUC, and PR AUC. Precision measures the percentage of positive predictions that are positive. The recall rate indicates the percentage of positive predictions out of all positive cases. The F1 score is the harmonic mean of precision and recall rate. AUC, the Area Under the Receiver Operating Characteristic Curve, assesses how well the model can distinguish between positive and negative classes. PR AUC, Area Under the Precision-Recall Curve, evaluates the model's performance on positive predictions in imbalanced datasets. Combining these metrics is considered to be a relevant way to evaluate a model's generalization ability on unseen data.

```
5 class ResNet50WithExtraBN(nn.Module):
6     def __init__(self, num_classes=len(dataset.classes)):
7         super(ResNet50WithExtraBN, self).__init__()
8         self.resnet = models.resnet50(pretrained=True)
9         self.resnet.fc = nn.Linear(self.resnet.fc.in_features, num_classes)
10
11        self.extra_bn = nn.BatchNorm1d(num_classes)
12
13    def forward(self, x):
14        x = self.resnet(x)
15        x = self.extra_bn(x)
16        return x
```

Figure 6: Python ResNet50WithExtraBN class used to add a batch normalization layer to a Resnet50 or Resnet152 model

Table 1. Final results of the models

	ResNet50	ResNet152
Valid. Acc.	85%	88%
Test Acc.	83%	86%
Precision	0.85	0.86
Recall	0.87	0.87
F1	0.85	0.86
AUC	0.99	0.99
PR AUC	0.92	0.93

The algorithms and training environments used to train the two ResNet models in this research were identical. The performance of both models is shown in Table 1. Both models reached a validation and test accuracy of > 80% along with a precision, recall rate, and F1 scores of > 0.85.

Compared to the two studies that used the same Kaggle "Best Artworks of All Time" dataset mentioned in Section 2, this work used the entire Kaggle dataset instead of only artworks from three artistic styles as done by Protik et al. and five artists as done by Gao et al., largely increasing the training difficulty of the model. At the same time the two models proposed in this work outperformed the models constructed by Protik et al. and Gao et al. (with validation set accuracies of 71% and 82% respectively) despite using a more comprehensive dataset, showing the potential of ResNet models with an added batch normalization layer to generalize on complex and large artwork datasets.

Compared with studies by Menis-Mastromichalakis et al. and Del Chiaro et al. which were based on ResNet models that also used comprehensive datasets, the models proposed in this work not only have a simpler structure and less computational requirements, but also have a better generalization ability (the models by Menis-Mastromichalakis et al. and Del Chiaro et al. resulted in validation set accuracies of 73% and 77% respectively).

Execution speed

When classifying a single artwork, both the ResNet50 and ResNet152 model returned an execution time < 0.03 seconds. These models were hence very efficient and suitable for real-time applications, such as when incorporated in an art recognition app. This speed enabled the models to provide instant results, enhancing user experience, especially when used on mobile devices or in situations where fast response times are important.

Results of classifying pieces from unfamiliar (unseen) artists from outside the dataset

After feeding 84 artworks from 24 artists that were not included in the dataset into the model, some intriguing patterns were found (see Appendix 1 for more details). For example, all three pieces by John Leech were classified as works of Vincent van Gogh. This likely occurred due to stylistic and structural

similarities between the two, as both artists were prominent figures in the realism art movement during the 19th century. Similarly, the three artworks by John McLaughlin were all classified as works of Kazimir Malevich, reflecting their shared late 19th-century origins and contributions to abstract art. These observations suggest that the algorithm emphasizes stylistic attributes over individual artist characteristics. Consequently, the pieces of some artists were classified as works of different artists from the same era who explored similar genres.

To improve classification accuracy, the model should prioritize style and genre over specific characteristics of individual artists or be trained on a dataset where classes only contain one genre of the artist's pieces. Emphasizing individual characteristics could hinder the model's generalization abilities since many artists create pieces across different art movements and genres. By transitioning to a style recognition model, where classes represent various genres and movements, the model could achieve more precise classification, as the classes would share greater stylistic similarities. Regardless, the classification of artworks using ML models relies significantly on the number of paintings in the training dataset, and demonstrates that most literature reporting high accuracy values in this field is largely meaningless unless a training dataset is used that incorporates (almost) all the paintings ever painted. Transfer learning is also likely to be unsuccessful in this regard due to the unique characteristics of paintings as opposed to photographs.

5 Areas of Improvement

Size of images in the original dataset

One limitation in the dataset used in this research is that the artwork images come in different sizes. Images having different sizes decrease the accuracy of the model because neural networks require the inputs to be of consistent size. When the images vary in size, they will have missing features and inconsistent representations after being input into the network. This inconsistency decreases the model's ability to learn and generalize patterns within the original images, resulting in poor model accuracy. Even though the model resized all the images in the dataset to 224 * 224 pixels before training, the proportions of many images changed after resizing them. This change caused the model to fail to learn the original structure of the image, causing overfitting in the training dataset. Hence, selecting a dataset with all images of the same size is crucial to ensure that the model performs to its greatest potential.

Difference in amount of images for each category in the original dataset

The amount of artwork images in each category in the dataset used in this work was different, which could decrease the accuracy of the model due to class imbalance. This imbalance causes the model to be biased towards the categories with more images, as it learns from them more often during training tasks. Consequently, the model performes well on the larger classes but

poorly on the smaller classes, resulting in low overall accuracy. To address this issue, techniques such as oversampling and weighted loss functions are often employed to balance the representation of each category. However, these algorithms both did not improve the performance of the two models used in this research, which may be due to the large difference in class size as well as the style and characteristics of artworks belonging to the same class. If the dataset used in this research had a more equal class size, then the models would have resulted in a greater prediction accuracy.

Lack of comparison with other networks

Only two models were experimented in this research, ResNet50 and Resnet152. There are many other models that can be explored, for example VGG models, Vision Transfer models, Inception models, AlexNet etc. These models may perform better than ResNet models on the Kaggle dataset and training environment.

6 Conclusion

In this work, artwork recognition models based on ResNet50 and ResNet152 with an added batch normalization layer were proposed. Utilizing many machine learning algorithms, the models could generalize on unseen images in diverse classes in the dataset, demonstrating the potential of ResNet models without added complicated architectures. In this paper, many methods for mitigating overfitting were also explored, such as schedulers, optimizers, data augmentation, and batch normalization, demonstrating their potential to increase model performance. The Resnet152 network proposed in this work achieved a validation set accuracy of 88% and a test set accuracy of 86%. With this innovative ResNet152 network, this work hopes to contribute to the field of art academically or culture heritage-wise.

7 References

1. Nag, P., Sangskriti, S., Jannat, M. (2020). A Closer Look into Paintings' Style Using Convolutional Neural Network with Transfer Learning. In *Algorithms for intelligent systems* (pp. 317–328). https://doi.org/10.1007/978-981-15-3607-6_26

2. Gao, J., Zhou, H., Zhang, Y. (2020). The Performance of Two CNN Methods in Artworks Aesthetic Feature Recognition. *2020 12th International Conference on Machine Learning and Computing*, 289–296. https://doi.org/10.1145/3383972.3383974

3. Menis-Mastromichalakis, O., Sofou, N., Stamou, G. (2020). Deep Ensemble Art Style Recognition. *2020 International Joint Conference on Neural Networks*. https://doi.org/10.1109/ijcnn48605.2020.9207645

4. Del Chiaro, R., Bagdanov, A., Del Bimbo, A. (2019). NoisyArt: A Dataset for Webly-supervised Artwork Recognition. *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, *4*, 467–475. https://doi.org/10.5220/0007392704670475

5. *Best Artworks of All Time*. (2019, March 2). Kaggle. https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time/data

6. Hassan, M. U. (2023, July 3). *ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks*. Neurohive / Neural Networks. https://neurohive.io/en/popular-networks/resnet/

7. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.90

8. Kingma, D. P., Ba, J. L. (2014). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*. https://doi.org/10.48550/arxiv.1412.6980

9. Loshchilov, I., Hutter, F. (2017). Decoupled weight decay regularization. *ICLR 2019*. https://doi.org/10.48550/arxiv.1711.05101

10. Tieleman, T., Hinton, G. (2012). Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. University of Toronto, Technical Report, 6.

11. Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1705.08292

12. *torch.optim — PyTorch 2.4 documentation*. (n.d.). PyTorch. https://pytorch.org/docs/stable/optim.html

13. Li, K. (2023, August 9). *How to choose a learning rate scheduler for neural networks*. neptune.ai. https://neptune.ai/blog/how-to-choose-a-learning-rate-scheduler

14. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning (ICML) in 2015*. https://doi.org/10.48550/arxiv.1502.03167

8 Appendix 1. Testing the models on Artists not included in the Kaggle dataset

Artwork Name	Actual Artist	Predicted Artist
The Sitting Room	Frances Hodgkins	Vincent van Gogh
A Barn in Provence	Frances Hodgkins	Vincent van Gogh
Der Totentanz Von Anno Neun	Albin Egger-Lienz	Pablo Picasso
Der Bauer	Albin Egger-Lienz	Henri Matisse
All Lanes of Lilac Evening	Max Ernst	Joan Miro
Appius Claudius Punished By The People	John Leech	Vincent van Gogh
Ariadne	George Frederick Watts	Titian
Austrian Family. The Mother	Josef Kriehuber	Gustav Klimt
Balance	Francis Picabia	Rene Magritte
Cabeza Azteca	Roberto Montenegro	Pablo Picasso
Christus Am Ölberg Und Die Marien Am Grabe	Lorenzo Monaco	Giotto di Bondone
City Types	Mstislav Dobuzhinsky	Henri Matisse
Creation of Eve	George Frederick Watts	Peter Paul Rubens
Dance at the Spring	Francis Picabia	Pablo Picasso
Daphne	George Frederick Watts	Titian
De-icing Aircraft	Eric Ravilious	Salvador Dali
The Blind	Albin Egger-Lienz	Pablo Picasso
Flea	Sue Coe	Marc Chagall
Arrangement of Jugs	Frances Hodgkins	Marc Chagall
Primulas	Frances Hodgkins	Albrecht Durer
Seated Women	Frances Hodgkins	Albrecht Durer
Ghosts of Departed Usurers	John Leech	Vincent van Gogh
Harvesters Resting	Jean-Francois Millet	Francisco Goya
Honor Dance	Kent Monkman	Vincent van Gogh
Gleaners	Jean-Francois Millet	Francisco Goya
Johann Strauss II	Josef Kriehuber	Pablo Picasso
Jonah	George Frederick Watts	Sandro Botticelli
Children	Josef Kriehuber	Francisco Goya
La Cosecha	Saturnino Herran	Vincent van Gogh
La Criolla del manton	Saturnino Herran	Francisco Goya
Lysanders	Eric Ravilious	Vincent van Gogh
Machine Turn Quickly	Francis Picabia	Pablo Picasso
Majorcan Fisherman	Roberto Montenegro	Frida Kahlo
Mammon	George Frederick Watts	Pierre-Auguste Renoir
Market in the Normandy	Theodore Rousseau	Vincent van Gogh
Marley's Ghost	John Leech	Vincent van Gogh
Mill	Sue Coe	Pablo Picasso
Miss Chief's Wet Dream	Kent Monkman	Vincent van Gogh

Morning	Theodore Rousseau	Alfred Sisley
Nation to Nation	Kent Monkman	Sandro Botticelli
New York Rooftops	Mstislav Dobuzhinsky	Vincent van Gogh
Miner's Wives	Ben Shahn	Pierre-Auguste Renoir
Second Allegory	Ben Shahn	Andy Warhol
Third Allegory	Ben Shahn	Henri Matisse
Number 13	John McLaughlin	Kazimir Malevich
Number 14	John McLaughlin	Kazimir Malevich
Number 6	John McLaughlin	Kazimir Malevich
October Idyll	Mstislav Dobuzhinsky	Henri Matisse
Oedipus Taken Down from the Tree	Jean-Francois Millet	Peter Paul Rubens
Badende in Braun	Otto Muller	Pablo Picasso
Der Mord	Otto Muller	Marc Chagall
Three Nudes	Otto Muller	Henri Matisse
Pink Roses in Brandy Glass	Frank Mason	Pierre-Auguste Renoir
Red Forest	Max Ernst	Rene Magritte
Richard Pionk President Salmagundi Art Club	Frank Mason	Francisco Goya
Silence of the Evening	Ferdinand Hodler	Amedeo Modigliani
Spring	Theodore Rousseau	Alfred Sisley
The Ancient of Days	William Blake	Frida Kahlo
The Augustan Bridge at Narni	Camille Corot	Peter Paul Rubens
The Centurion	Frank Mason	Francisco Goya
The Elephant Celebes	Max Ernst	Giotto di Bondone
The Forest of Fontainebleau	Theodore Rousseau	Vincent van Gogh
The Lover's Whirlwind	William Blake	Rene Magritte
The Oxen	Frank Mason	Rene Magritte
The Raft of Medusa	Théodore Géricault	Peter Paul Rubens
The Sower	Jean-Francois Millet	Titian
The Vale of the White Horse	Eric Ravilious	Salvador Dali
The Wild Horse Race at Rome	Théodore Géricault	Peter Paul Rubens
The Woodman	Ferdinand Hodler	Vincent van Gogh
The Wounded Curirassier	Theodore Géricault	Vincent van Gogh
Tree in the Workshop Garden	Ferdinand Hodler	Vincent van Gogh
Triumph	Sue Coe	Pieter Bruegel
Twilight in Sologne	Theodore Rousseau	Vincent van Gogh
Two Women with Poppies	Francis Picabia	Rene Magritte
Ubu Imperator	Max Ernst	Rene Magritte
Venice Gondola on Grand Canal	Camille Corot	Vincent van Gogh
Virgin and Child with Six Angels	Lorenzo Monaco	Giotto di Bondone
Visions de los Vencidos	Roberto Montenegro	Henri Matisse
Voltarra the Citadel	Camille Corot	Vincent van Gogh
Vox Angelica	Max Ernst	Paul Klee
Water Walker	Kent Monkman	Francisco Goya
Woman Reading in a Landscape	Camille Corot	Titian
Women and Bulldog	Francis Picabia	Frida Kahlo