IT461: Practical Machine Learning Project Report

Artsy

Supervised by Luluah Alhussain

Prepared by

Meshael Alessa, 442201371

Emtenan Alghamdi, 442200829

Maha Almuaythir, 442201336

Jomanah Alomar, 442201379

Table of Contents

Introduction: *3*
Related Works: 4
Data: 5
Methods: 6
Experiment: 8
Results and Discussion: 11
Conclusion: 16
Contributions: 16
References: 17

Table of Figures

Figure 1 how does Artsy model work *4*
Figure 2 Surreal symphonies dataset example *5*
Figure 3 Surreal symphonies dataset example *5*
Figure 4 WikiArt dataset example *6*
Figure 5 WikiArt dataset example *6*
Figure 6 Preprocessing and normalization. *9*
Figure 7 Feature extraction. *10*
Figure 8 Model initialization. *10*
Figure 9 Training loop. *10*
Figure 10 Evaluation. *10*
Figure 11 The models learning curves. *12*
Figure 12 Confusion Matrices. *12*
Figure 13 Confusion Matrices for the second dataset. *14*
Figure 14 The models learning curves for the second dataset. *15*

Introduction:

In the intersection of art and technology, we embark on a venture to harness the power of computer vision to classify paintings into their respective art styles. This endeavor is not just a technical challenge; it holds profound implications for how we understand, preserve, and interact with cultural heritage.

Background:

Art classification by style is a pivotal task in various domains such as digital humanities, art conservation, and education. It's a complex problem due to the subjective nature of art perception and the wide variety of art styles. With the advancements in machine learning, especially in image processing, there is a significant opportunity to automate this task, making it more efficient and accessible.

Problem Statement:

The core challenge we address is the automatic categorization of paintings by art style using machine learning models. We will be using two distinct datasets: kaggle dataset and WikiArt dataset. Our objective is to assess how the same models perform with datasets of varying quality, which will shed light on the robustness of our system in art style detection.

Intended Task Illustrated:

The process is as follows:

Input: A digital image of a painting, accompanied by two key data representations:

Histogram of Oriented Gradients (HOG) features that capture the painting's texture and edge information.

Color histogram that encapsulates the color distribution of the artwork.

Output: The output is a categorical label, assigned by our model, identifying the painting's art style from a diverse set of possibilities.

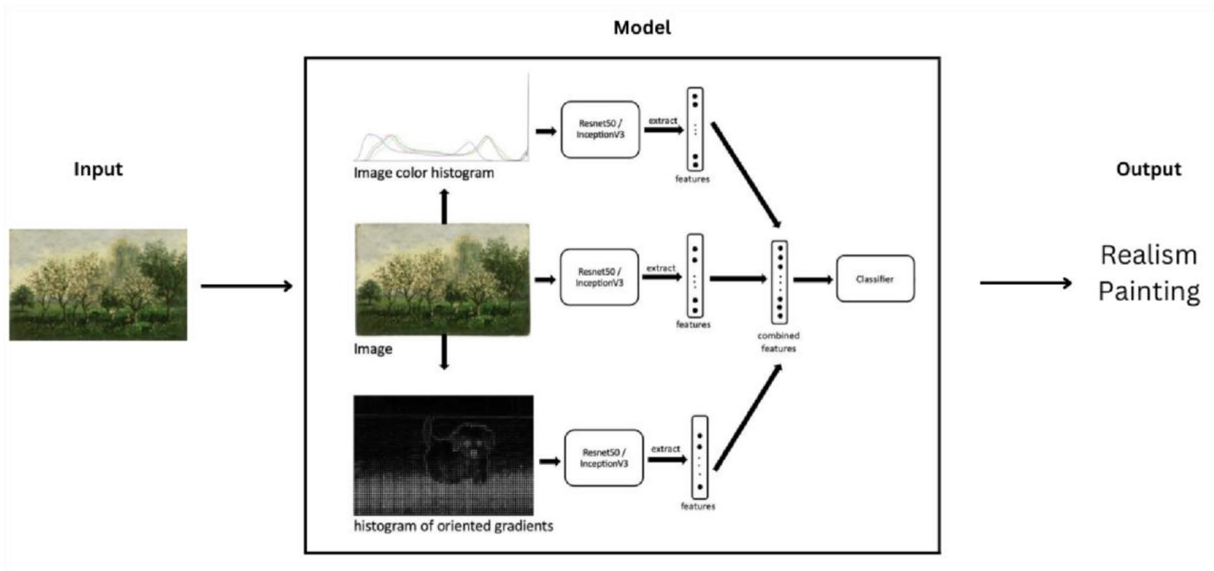


Figure 1 how does Artsy model work

Related Works:

1. Introduction to Art Style Detection

The integration of AI in detection of art styles and movements has long been a topic of interest in the realms of computer vision and deep learning. Over the years, researchers have delved into the potential of deep learning algorithms to find patterns and themes for classification and generation tasks. In this context, we'll review four papers related to the topic art style detection, shedding light on the algorithms and methodologies they employ. Subsequently, we will explain our unique approach to this challenge and highlight the novel contributions we bring to the domain.

2. Related works discussion

In the field of art style recognition using deep learning, several studies have demonstrated impressive results. The RASTA classifier, utilizing the Wikipaintings dataset with 80,000 images and fine-tuning the last 20% layers of the ResNet50 model, achieved a 62% accuracy. This was further enhanced by techniques like bagging and data augmentation [1]. Another study employing Vision Transformers and MLP Mixer on the WikiArt dataset reported nearly 40% accuracy after 100 training epochs [2]. A different approach involved a two-channel ResNet50 architecture, one for RGB images and another for brush stroke data, reducing the error rate on various WikiArt datasets from 49.91% to 47.2% [3]. Finally, a two-stage deep learning method, first using a deep CNN on image patches and then a neural network on the first stage's results, consistently outperformed other methods, reaching a peak accuracy of 77.53%

with the InceptionV3 model. These studies highlight the diverse methodologies and their effectiveness in classifying fine art styles [4].

Data:

Our research utilizes datasets comprising artistic imagery from five distinct classes, namely Abstract Expressionism, Art Nouveau Modern, Fauvism, Minimalism, and Rococo. The objective is to evaluate our module's proficiency in adapting to variations in balance and assimilating diverse art styles.

Sourced from Kaggle [5] and Wikiart [5], The Kaggle dataset includes approximately 1500 images, while the Wikiart dataset is more comprehensive with about 5000 images. Both datasets are pre-organized and labeled according to their respective categories as shown in Table1, eliminating the need for further data preprocessing.

Dataset name	Number of classes	Number of examples	Example
Surreal Symphonies (A Dataset of Diverse Art) [6]	This dataset contains 5 classes: Abstract Expressionism, Art Nouveau Modern, Fauvism, Minimalism, and post Impressionism.	The number of images for each category are approximately 300 per style.	Fauvism art style:  *Figure 2 Surreal symphonies dataset example* Art Nouveau art style:  *Figure 3 Surreal symphonies dataset example*

WikiArt Dataset (Refined) [5]	This dataset contains 5 classes: Abstract Expressionism, Art Nouveau Modern, Fauvism, Minimalism, and Rococo.	In WikiArt the pictures are labeled with numbers corresponding to certain art style, this means the dataset is ready to be used and perfect for our module.	Rococo:  *Figure 4 WikiArt dataset example* Abstract Expressionism:  *Figure 5 WikiArt dataset example*

Table 1 Datasets general information and examples.

Methods:

Machine Learning Models Employed:

The primary algorithm employed in this project is a deep neural network architecture known as ResNet50, a variant of the Residual Network architecture that is 50 layers deep. ResNet50 is widely recognized for its effectiveness in image recognition tasks due to its deep architecture and skip connections, which mitigate the vanishing gradient problem allowing for deeper model training without performance degradation. Additionally, we leveraged Histogram of Oriented Gradients (HOG) and color histograms as part of our feature engineering process to capture texture and color distribution, which are significant attributes in art style classification.

Justification of Methods:

ResNet50 was chosen for its state-of-the-art performance on image classification tasks and the ability to leverage transfer learning from pre-trained models. This choice was pertinent considering the complexity of art style classification, where styles can be nuanced and not easily separable through simple features. The skip connections in ResNet allow for training deeper networks by addressing the degradation problem, hence providing a robust framework for feature extraction.

HOG features were included because they are effective at capturing the shape and texture information by considering the distribution of local gradients and edge directions, which are important factors in distinguishing between different art styles. Color histograms complement HOG by capturing the color profile of the artwork, which is another important stylistic feature.

Preprocessing and Feature Engineering:

The preprocessing pipeline consisted of resizing images to a uniform size of 224x224 pixels, which is a standard input size for ResNet50, and normalizing the images to the range [0,1] to facilitate model convergence. The dataset images were converted to tensors to be compatible with PyTorch's computation graph. In feature engineering, we extracted HOG features to encapsulate the local shape and texture information, and color histograms to capture the color distribution within the artworks. These features were then concatenated with the output of the penultimate layer of the ResNet50 model (before the classification layer), resulting in a rich feature set combining shape, texture, and color characteristics.

Model Training:

The neural network classifier, termed "DeepNN," was custom-built with fully connected layers. The input dimension was set to accommodate the concatenated feature set from ResNet50, HOG, and color histograms. Dropout was applied as a regularization technique to prevent overfitting. The model was trained with a cross-entropy loss function, which is appropriate for multi-class classification problems. Different configurations of the model were trained with variations in the number of hidden layers, dropout rates, and learning rates, and optimization was performed using Adam optimizer due to its adaptive learning rate capabilities, which often leads to faster convergence.

Model Evaluation:

Performance evaluation was conducted using a validation set during training to fine-tune hyperparameters and avoid overfitting. Post-training, a testing phase was employed using unseen data to assess the generalization capability of the model. The confusion matrix was used as the primary tool for performance evaluation, allowing for a detailed analysis of the model's predictive capabilities across different classes.

In summary, the combination of a deep learning model with traditional computer vision features encapsulated a comprehensive approach to address the complex problem of art style classification. The systematic methodology of preprocessing, feature engineering, model architecture design, and rigorous evaluation led to a robust classification system.

Experiment:

Architecture and Structure of the Models:

ResNet50: The core architecture employed in our experiments is ResNet50, a 50-layer deep convolutional neural network with residual connections. These connections allow the network to skip one or more layers, which prevents the vanishing gradient problem and enables the network to learn an identity function at the very least, ensuring that deeper networks do not result in higher training error. The key components of ResNet50 are convolutional layers, batch normalization, ReLU activations, and pooling layers. The final layer is a fully connected layer that outputs the probabilities of each class. **DeepNN:** The custom Deep Neural Network (DeepNN) used for classification consists of several fully connected layers with ReLU activation functions and a final softmax layer. Dropout regularization was included to prevent overfitting by randomly setting a subset of activations to zero during training. This network took the concatenated feature vectors as input, which combined the outputs of ResNet50, HOG, and color histogram features.

Training Process

Preprocessing and Feature Engineering:

- The images were resized to 224x224 pixels, the standard input size for ResNet50.
- Normalization was applied to scale pixel values to the range [0,1].
- HOG features were extracted to capture edge and gradient structure indicative of artistic style.
- Color histograms were computed to summarize the color distribution of the images.
- These features were then concatenated to form a comprehensive feature vector for training.

Dataset Splitting:

The dataset was split into training (80%), validation (10%), and testing (10%) sets to ensure a robust evaluation. This split was chosen to provide a large enough training set while still having sufficient data for model validation and testing.

Regularization:

Dropout was the primary regularization technique used, with rates experimented between 0.3 to 0.5 to mitigate overfitting.

Hyperparameter Tuning:

Hyperparameters tuned include the learning rate, number of hidden layers, number of neurons in each layer, and dropout rate.

A combination of manual tuning and empirical observation was used, informed by validation set performance.

Learning rates explored ranged from 0.0001 to 0.001, dropout rates from 0.3 to 0.5, and different configurations for the number of hidden layers and neurons.

The optimal values were selected based on the highest accuracy and lowest loss on the validation set.

Model Evaluation:

The model's performance was evaluated using the accuracy metric and the confusion matrix. The confusion matrix provided insight into the classification performance across different classes, identifying any biases or weaknesses in the model.

Computational Resources:

The training was conducted on a Tesla T4 GPU due to the computational demands of deep learning models like ResNet50.

Libraries such as PyTorch were used for model implementation, training, and evaluation.

Code Snippets:

```
[ ]  # Define transforms for the images
     transform = transforms.Compose([
         transforms.Resize((224, 224)),
         transforms.ToTensor(),
     ])
```

Figure 6 Preprocessing and normalization.

```
class FeatureExtractor:
    def __init__(self):
        self.resnet50 = models.resnet50(pretrained=True)
        self.resnet50 = torch.nn.Sequential(*list(self.resnet50.children())[:-1])
        self.resnet50.eval()
```

```
    def extract_all_features(self, image_tensor):
        image_np = self.tensor_to_numpy(image_tensor)
        hog_features = self.extract_hog(image_np)
        color_hist_features = self.extract_color_histogram(image_np)
        resnet_features = self.extract_resnet50(image_tensor)
        return np.concatenate([hog_features, color_hist_features, resnet_features])
```

Figure 7 Feature extraction.

```
# Initialize the model
model = DeepNN(input_dim, hidden_dims, output_dim, dropout_rate)
```

Figure 8 Model initialization.

```
for epoch in range(num_epochs):
    # Training
    model.train()

    train_loss_epoch = 0.0
    train_correct = 0

    train_loop = tqdm(enumerate(train_loader), total=len(train_loader), leave=True)
    for batch_idx, (data, target) in train_loop:
        optimizer.zero_grad()
        output = model(data)
        loss = criterion(output, target)
        loss.backward()
        optimizer.step()
```

Figure 9 Training loop.

```
# Validation
model.eval()
valid_loss = 0.0
correct = 0
valid_loop = tqdm(valid_loader, leave=True)
with torch.no_grad():
    for data, target in valid_loop:
        output = model(data)
```

Figure 10 Evaluation.

In summary, the training process was comprehensive, involving feature extraction, careful preprocessing, and a structured approach to training deep learning models with hyperparameter tuning and performance evaluation.

Results and Discussion:

The performance of the implemented machine learning models was evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score. These metrics give a comprehensive understanding of the models' capabilities in classifying different art styles.

Performance Metrics

Model/Features	Accuracy	Precision	Recall	F1-Score
ResNet50 Only	83.11%	0.8355322223783244	0.8310679611650486	0.8301111794663596
ResNet50 + HOG	86.02%	0.8347637964084692	0.8330097087378641	0.8307627384810377
ResNet50 + Color Histograms	56.89%	0.5715940202671594	0.5689320388349515	0.5575683850923044
ResNet50 + HOG + Color Hist.	65.63%	0.6656879071976715	0.6563106796116505	0.6591548180642341

After experimenting, we noticed that color histograms cause a much worst performance when added to the features, while histograms of oriented gradients help the performance by a slight increase of nearly 4%.

Visualizations and Performance Understanding

The provided visual data presents the test accuracies and learning curves for the different models, along with their corresponding confusion matrices. These visualizations offer a clear comparison of the performance between the four variants of the implemented model.

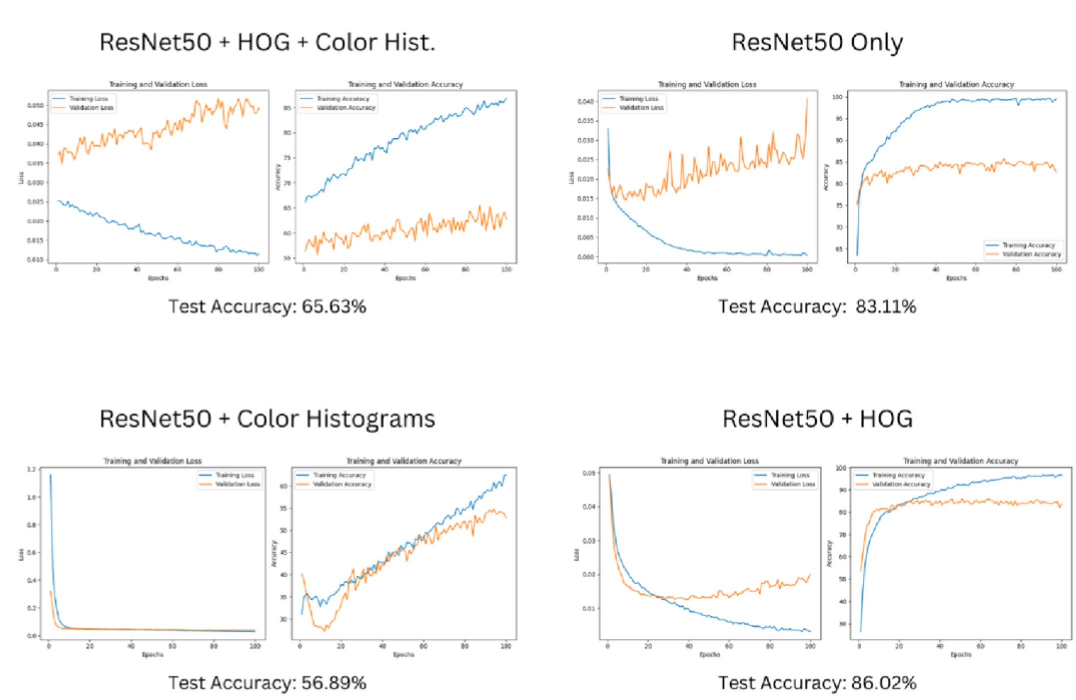


Figure 11 The models learning curves.

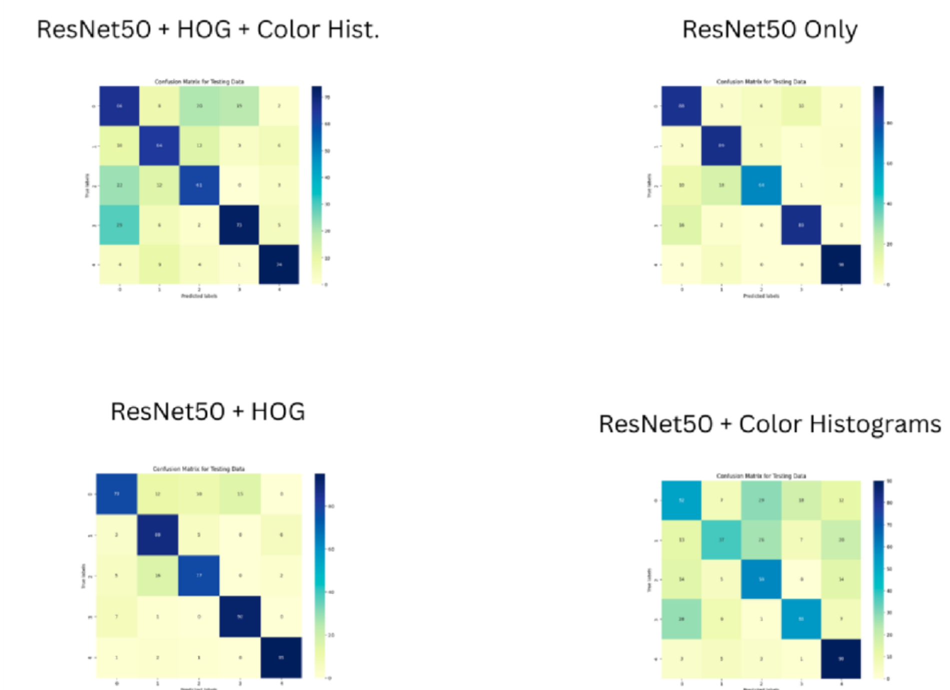


Figure 12 Confusion Matrices.

Learning Curves and Model Generalization

The learning curves as shown in figure11 indicate that the ResNet50 Only and ResNet50 + HOG models exhibit stable convergence of training and validation losses, suggesting good generalization without significant overfitting. On the other hand, the ResNet50 + Color Histograms model shows a disparity between training and validation performance, indicating potential overfitting to the training data, which is also reflected in the lowest test accuracy. The ResNet50 + HOG + Color Histograms model, while better than the color histogram-augmented model alone, still underperforms compared to the HOG-enhanced and baseline models.

Confusion Matrices Insights

The confusion matrices shown in figure12 reveal that the ResNet50 + HOG model achieves a more balanced classification across different classes, as indicated by higher diagonal values representing correct classifications. The ResNet50 Only model, while generally performing well, shows confusion between certain classes. The models incorporating color histograms, particularly when used without HOG features, appear to struggle with class differentiation, as indicated by the more even distribution of predictions across incorrect classes.

Comparative Analysis

The superior performance of the ResNet50 + HOG model suggests that HOG features are highly informative for art style classification, capturing essential textural information that is not encoded in the raw pixel values processed by ResNet50. The underperformance of the color histogram-augmented models may indicate that color alone is not as discriminative for the styles in the dataset or that the method of integrating color histograms needs refinement.

Interestingly, the combination of all features (ResNet50 + HOG + Color Histograms) does not yield the best results, possibly due to the curse of dimensionality or the introduction of noise through the color histograms, which could dilute the effectiveness of the more informative HOG features.

Interpretation of Results

The results demonstrate the complexities of feature interaction within machine learning models for art classification. They suggest that while deep learning models can extract high-level abstractions, the integration of traditional features such as HOG can be beneficial. However, the combination of features must be carefully considered to ensure complementary rather than conflicting information is provided to the model.

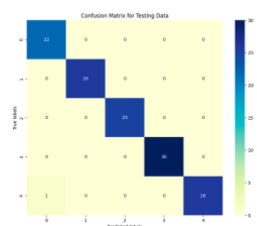
In conclusion, the analysis underscores the importance of feature selection in machine learning. It suggests that while deep learning provides a powerful tool for automatic feature extraction, traditional feature engineering techniques still hold value and can significantly impact model performance when appropriately applied. The choice and integration of features remain a critical aspect of model design and should be guided by both empirical results and domain knowledge.

For the second dataset (Surreal art dataset)

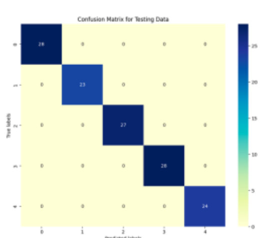
Model/Features	Accuracy	Precision	Recall	F1-Score
ResNet50 Only	100%	1.0	1.0	1.0
ResNet50 + HOG	100%	1.0	1.0	1.0
ResNet50 + Color Histograms	100%	1.0	1.0	1.0
ResNet50 + HOG + Color Hist.	100%	1.0	1.0	1.0

ResNet50 + HOG + Color Hist.

ResNet50 Only



ResNet50 + HOG



ResNet50 + Color Histograms

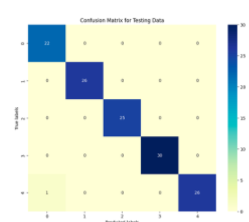


Figure 13 Confusion Matrices for the second dataset.

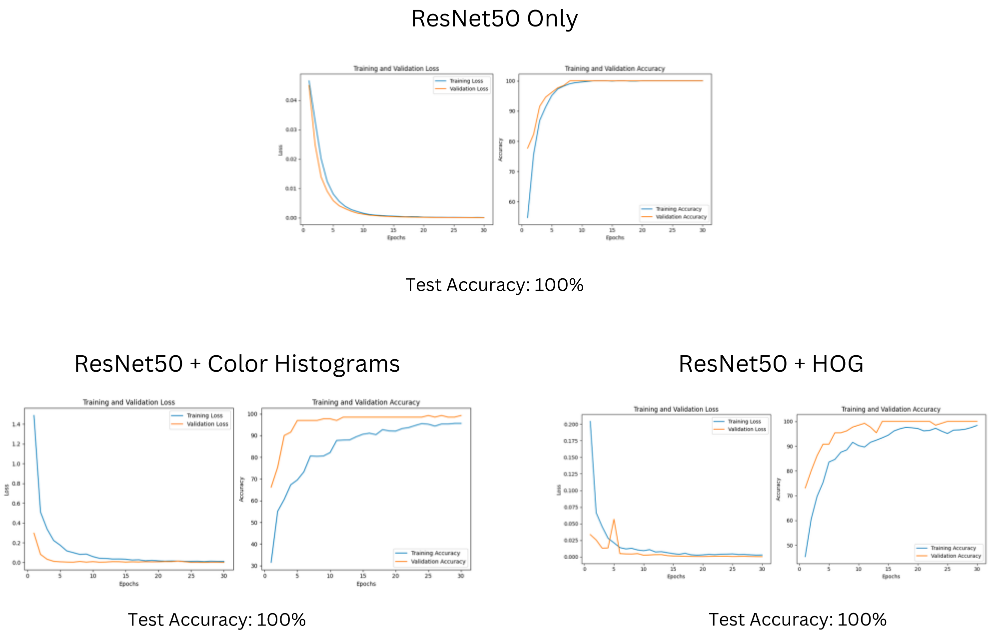


Figure 14 The models learning curves for the second dataset.

The uniform perfection in model performance on the second dataset necessitates a discerning analysis. While the results could be misinterpreted as evidence of the models' superior predictive power, such impeccable metrics are rarely encountered in real-world datasets, which are inherently more diverse and complex. The consistent scores across different models may indicate an oversimplified dataset, raising concerns about the models' ability to generalize beyond this specific context. This situation underscores the critical role of dataset diversity in truly assessing and refining the capabilities of machine learning models, a key consideration for future algorithm development and evaluation.

Conclusion:

This project's exploration into art style classification revealed that integrating deep learning with traditional image processing techniques, specifically ResNet50 with Histogram of Oriented Gradients (HOG), could yield significant performance gains. The best model achieved an 86.02% accuracy, underscoring the importance of texture features in art analysis. Challenges included balancing feature types and combating overfitting, which presents opportunities for future work. Potential improvements involve refined feature integration, advanced regularization, data augmentation, ensemble methods, exploring different architectures, and expanding the dataset. The findings affirm the value of combining classical and contemporary approaches in machine learning to enhance model efficacy. Challenges encountered during the project include the integration and balancing of feature types, which proved to be non-trivial. The addition of color histograms introduced complexity without a corresponding increase in performance, suggesting that not all features harmonizing to produce a better model. Another limitation was the potential for overfitting when using a rich feature set, which was mitigated to some extent by dropout regularization but remained a concern.

At the end, the project successfully demonstrated the synergy between convolutional neural networks and traditional feature extraction methods. It provided valuable insights into the challenges of feature integration and model complexity, laying the groundwork for future research in the domain of art classification with machine learning.

Contributions:

Name	Contribution
Jomanah Alomar	Introduction - Related Works - Data
Meshael Alessa	Results and Discussion
Emtenan Alghamdi	Experiment
Maha Almuaythir	Methods - Conclusion

References:

[1] B. N. F. Y. Adrian Lecoutre, "Recognizing Art Style Automatically in painting," 2017.

[2] S. N. P. S. S. W. S. K. G. Lazaros Alexios Iliadis, "Artwork Style Recognition Using Vision Transformers and MLP Mixer," *technologies,* vol. 10, no. February 2022, 2022.

[3] S.-h. Z. a. Z. X. Xingsheng Huang, "Fine-Art Painting Classification via Two-Channel Deep Residual Network," Springer, Cham, 2017.

[4] E. P. a. M. L. CATHERINE SANDOVAL, "Two-Stage Deep Learning Approach to the," IEEE Access, 2019.

[5] W. R. a. C. C. S. a. A. H. a. T. K. Tan, "Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork," IEEE Transactions on Image Processing, 2019. [Online]. Available: https://doi.org/10.1109/TIP.2018.2866698. [Accessed 16 September 2023].

[6] "Surreal Symphonies (A Dataset Of Diverse Art)," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/cyanex1702/surreal-symphonies-a-dataset-of-diverse-art. [Accessed 17 September 2023].