# 4IM06 Article: Abstract vs Figurative Painting Classification


Télécom Paris

Francisco Nicolás Noya
noyabagnasco@telecom-paris.fr

Joel Cabrera
angel.cabreradechia@telecom-paris.fr

Pedro de Almeida Marim
pedro.almeida@telecom-paris.fr

Luciana Munhos
luciana.munhos@telecom-paris.fr


June 24, 2025


**Abstract**

This study addresses the task of classifying paintings into two categories: figurative and abstract. Due to the subjective and often ambiguous nature of these images there is not a clear border between the two, complicating automated classification. Using the ResNet-50 architecture pre-trained on ImageNet, we designed and compared three neural network variants: a baseline single-branch model, a two-branch model combining global and local image features, and an extended two-branch model with additional dense layers for feature fusion. Our best-performing model achieved a macro F1-score of 90%, demonstrating strong classification capability.


## 1 Introduction

The objective of this project is to classify paintings into two categories: figurative and abstract. Contrary to what one might expect, this task is far from trivial. The primary challenge lies in the lack of a universally accepted definition within the art world for what constitutes figurative or abstract art. The boundaries between the two are often ambiguous—what one may perceive as a figurative work might be interpreted by another as abstract. In many cases, only the artist themselves can definitively classify the piece.

Despite the inherently ill-posed nature of this problem, the AI community has shown considerable interest in developing models capable of classifying artworks across various styles and categories. In this study, we narrow our focus to the binary classification of paintings as either figurative or abstract, leaving the exploration of additional categories for future work.

Our approach is based on the ResNet-50 architecture, pre-trained on the ImageNet dataset. Using this model as a foundation, we developed three variants aimed at accurately classifying images into the two target categories.

Given the subjective nature of this classification task—heavily influenced by the annotator's perspective and interpretation of art—we introduced a secondary objective. In addition to assigning a class label, our model also outputs a probability score reflecting the degree of "abstractness" of each painting. This probabilistic output provides insight into the model's confidence. For instance, a score close to 0.5 indicates uncertainty, suggesting that the artwork resides near the conceptual boundary between figurative and abstract. This allows for a better understanding of both the image and the model's decision-making process.

# 2 Related Work

## 2.1 Deep Residual Learning for Image Recognition

ResNet (Residual Network), introduced in [1], transformed deep learning by enabling the training of very deep neural networks through residual learning. Instead of learning a direct mapping, each layer learns a residual function $F(x) = H(x) - x$, and the original input x is added back via a shortcut connection: $F(x)+x$. These skip connections help avoid the vanishing gradient problem, making it possible to train networks with a great number of layers, like the ones used in this work, which has 53 layers.

## 2.2 Resnet For Painting Classification

The idea of using a ResNet architecture for painting classification is not novel; it was first proposed in [2]. In that work, the authors argue that the object recognition capabilities of ResNet can be leveraged to distinguish various aspects of a painting that are relevant for classification, such as style and depicted objects. For instance, if the network detects a human face or recognizable figures, it is likely that the painting is figurative.

This motivated our decision to base our architecture on ResNet. In addition to its proven ability to extract semantically meaningful features from images, ResNet offers strong performance with minimal training effort (see Figure 1). These characteristics made it a natural and practical choice as the foundation for our models.

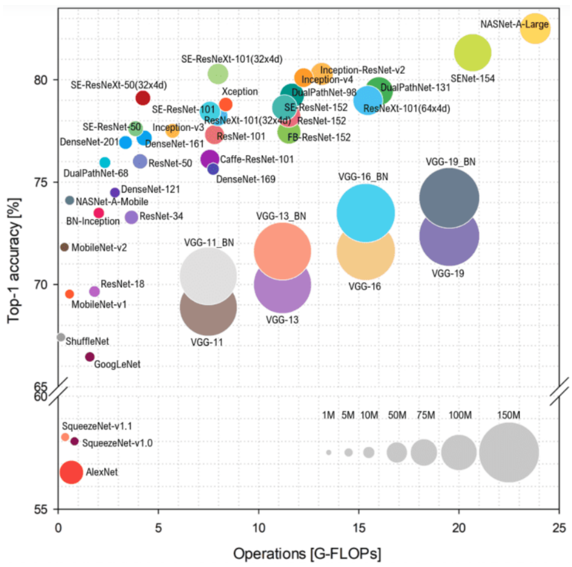



Figure 1: Computation vs Power of most common Computer Vision architectures. As we can see ResNet50 has a huge performance with little computation requirements.

## 2.3 Merging The Output Embeddings

Some of our methods involved merging the outputs of multiple ResNet-50 models. To enable this, a merging protocol had to be defined. In this work, we followed the strategy proposed in [3], where the embeddings from each model are simply concatenated to form a larger combined embedding. Specifically, the output of one ResNet-50 was concatenated with the output of another.

### 2.4 Grad-CAM

To further analyze the model's behavior, we explored which regions of the paintings were being considered when predicting each class. We employed Grad-CAM (Gradient-weighted Class Activation Mapping), a technique introduced in [4], to generate visual explanations of the network's decisions. Grad-CAM produces a heatmap superimposed on the input image, highlighting the areas that most strongly influenced the model's output. This visualization proved crucial for interpreting the model's internal reasoning and understanding how it distinguishes between abstract and figurative artworks.

## 3 Our Method

We implemented and compared three neural network architectures, all based on the ResNet-50 model pre-trained on ImageNet, to address the task of classifying paintings as abstract or figurative.

### 3.1 Baseline: Single-Branch ResNet

The first model serves as a baseline. It consists of a standard ResNet-50 to which we feed the entire painting image as input. The final fully connected layer is adapted to match the binary classification problem output. This model leverages the global structure and semantics present in the full image and has the advantage of being straightforward to train and evaluate.

### 3.2 Two-Branch Model

To capture finer stylistic details that may not be apparent in the entire image, we developed a second model inspired by the two-branch architecture proposed in [3]. In our design, one ResNet-50 processes the full image, while a second ResNet-50 processes a central patch cropped from the same image. The idea is to combine global and local representations, where the patch branch focuses on texture and brushstroke-level cues indicative of abstract or figurative styles.

The feature maps from both branches are flattened and concatenated to form a single embedding, which is then passed to a final linear layer for binary classification. This strategy allows the model to learn complementary features from two distinct perspectives of the same artwork, similar to how multi-resolution fusion is handled in remote sensing applications like [3], where PAN and MS data are processed separately and merged.

### 3.3 Extended Two-Branch Model with Dense Fusion

In the third and final model, we further extended the two-branch approach by adding multiple fully connected layers after the concatenation of the branch features. The rationale behind this enhancement is to allow more complex interactions between the global and local features before classification. Instead of relying on a single linear combination of the concatenated features, the added depth enables the model to learn hierarchical representations and refine decision boundaries.

This more expressive architecture aims to better capture the subtleties in the interplay between abstractness and figurativeness, particularly for ambiguous or borderline cases.

### 3.4 Fine-Tuning Strategy

Across all models, we employed transfer learning by initializing the ResNet-50 backbones with pre-trained ImageNet weights. To adapt the networks to our specific task while retaining general visual knowledge, we fine-tuned only the last 20% of the layers in each ResNet. The remaining layers were frozen during training.

# 4 Experiments

The original dataset contains a total of 7715 images corresponding to different paintings. 6400 of these paintings are figurative, while 1315 are abstract. As stated before, the main goal is to classify new paintings into one of the two classes.

The main problem encountered was the class imbalance: we have 5 times more figurative images than abstract ones. This is problematic since we could have a misleading sense of good accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall the accuracy formula in 1. For instance, if we implement a (very) simple classifier that predicts every painting as figurative, we would achieve an accuracy of 82%! To tackle this problem, we used a weighted loss to penalize more the class with fewer examples. In particular, we used the following weights:

$$w_f = \frac{N}{N_f} \quad w_a = \frac{N}{N_a}$$

where $N$ is the total number of images, and $N_f$ and $N_a$ are the numbers of figurative and abstract paintings, respectively. Therefore, the weights used are $w_f \approx 1.21$ and $w_a \approx 5.87$. This allows us to get a better feedback on the model's performance using the F1-Score defined in 2.

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{2}$$

where

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}$$

On the other hand, we normalized the images to keep their values in the range $[0, 1]$, and we also added zero-padding to obtain images of size $3 \times 224 \times 224$.

## 4.1 One-Branch ResNet50

We can observe the configuration of the first experiment using our model in Table 1. The results are presented in Table 2. We observe that, even though there is a clear overfitting (as shown by the 99% score on the training set), the model achieves very good results on the validation and test sets, with nearly 90% F1-score.

We can also observe the confusion matrix on the test set for this experiment (Figure 2). It seems that the classifier still has a slight bias toward predicting paintings as figurative, but the results are not too bad.

| | |
|---|---|
| **Model architecture** | ResNet50 |
| **Loss function** | CrossEntropyLoss (with class weights) |
| **Training epochs** | 50 |
| **Batch size** | 32 |
| **Learning rate** | 0.001 |
| **Data augmentation** | Enabled (standard) |
| **Scheduler** | Disabled |
| **Weighted loss** | Yes ($w_0 = 1.21$, $w_1 = 5.87$) |

Table 1: Configuration of the ResNet50 model (Experiment 1).

Table 2: Comparison of performance metrics across training, validation, and test sets (Experiment 1).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Accuracy | 0.9972 | 0.9287 | 0.9371 |
| F1-score | 0.9951 | 0.8751 | 0.8928 |
| Precision | 0.9925 | 0.8625 | 0.8759 |
| Recall | 0.9978 | 0.8894 | 0.9129 |

## 4.2 Level of abstraction

To assign a score indicating how abstract or figurative a painting is, we used the logits of the final layer. The predicted class is the one with the highest probability, and this number reflects the model's confidence. A value close to 0 indicates a strong belief that the image is figurative, while a score near 1 indicates high confidence in the image being abstract.

### 4.2.1 Correctly classified paintings

From the results in Figure 3, we can say that when the model correctly predicts that a painting is abstract, it is very confident (probabilities close to 1). However, when it correctly classifies figurative paintings, the model is generally less confident, with values starting from 0.15.

### 4.2.2 Incorrectly classified paintings

From Figure 4, we can draw several interesting observations. First, in 4b, note how the misclassified figurative paintings have scores close to the threshold (0.5), meaning the model was not very confident in predicting them as figurative. On the other hand, 4a shows abstract classifications for actually figurative images. At first glance, they appear abstract. It is only after a closer look that we notice some figures in the painting, revealing their figurative nature. We can say the model was misled by the painter's style, which introduced ambiguity the model couldn't handle.

Additionally, it's possible that some paintings in the dataset are mislabeled. Some labeled as figurative may actually be abstract. This could confuse the model during training, as it may be penalized for a correct prediction due to a labeling error. In such cases, the issue is not the model but rather the quality of the dataset.

We also wanted to explore optimization techniques to accelerate convergence and improve generalization. In particular, we implemented a learning rate scheduler using `ReduceLROnPlateau`, which monitors the validation F1 score and reduces the learning rate by a factor of 0.5 when no significant improvement (less than 0.1% relative gain) is observed for two consecutive validation steps. Figure 5 shows the effect of this strategy, comparing the validation F1 score over training epochs with and without the scheduler. We can observe that the scheduler leads to more stable and sometimes higher performance in later stages of training.

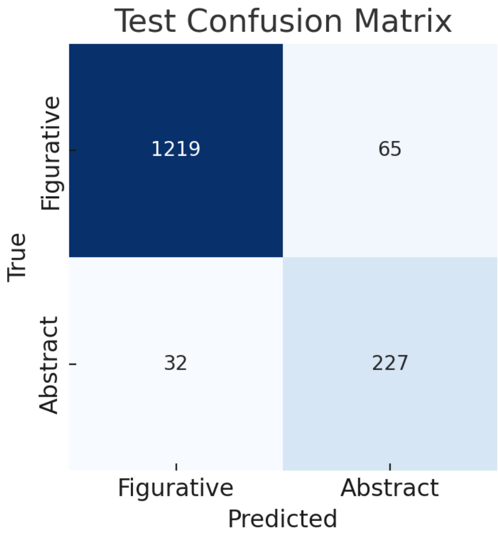
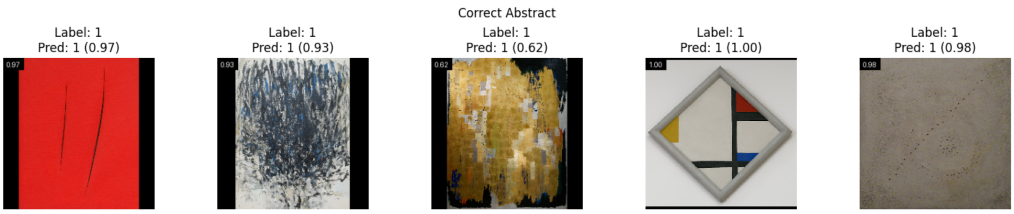
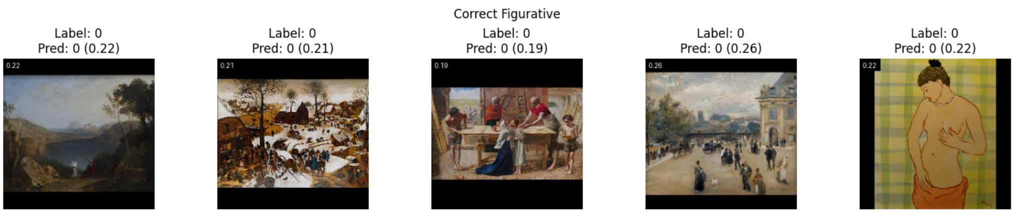



Figure 2: Confusion matrix for the first experiment




(a) Correctly classified abstract and figurative paintings. Model predictions match the ground-truth labels with high confidence.




(b) More examples of correct classifications. The model correctly distinguishes between abstract and figurative styles.

Figure 3: Examples of correctly classified paintings from both classes in the test set. Each image displays its label, prediction, and probability.

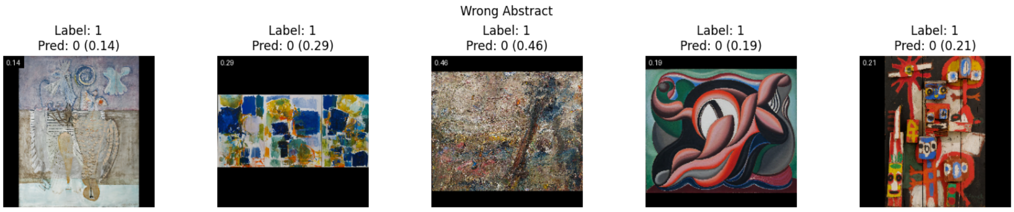
## 4.3 Two-Branch ResNet50 experiments

We conducted two experiments using a Two-Branch ResNet50 architecture. In both cases, we trained the model using standard data augmentation and weighted cross-entropy loss to address the class imbalance in the dataset. The goal was to assess the model's performance both under ideal conditions (clean images) and in the presence of noise, to evaluate robustness.

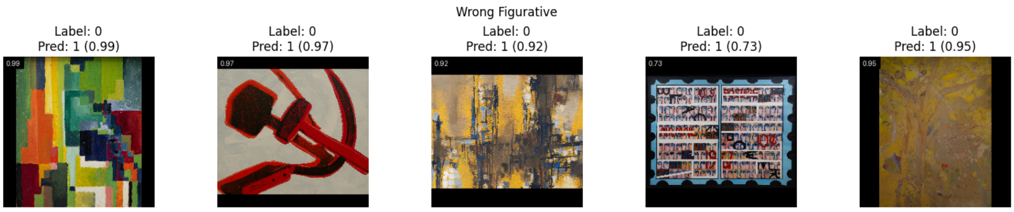
### 4.3.1 Experiment 2.1: Original Images

In the first setup, we trained the Two-Branch ResNet50 on the original dataset without any artificial corruption. The configuration used in this experiment is shown in Table 3.

The model achieved near-perfect performance on the training set, and although there was a noticeable drop in generalization, the test performance remained solid—with an accuracy of 0.8613 and an F1-score of 0.8035. This decline from the validation set highlights a slight overfitting, but

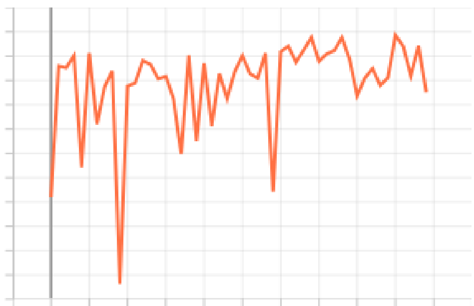



(a) Misclassified abstract paintings. Predictions tend toward 0.5, indicating model uncertainty in distinguishing abstract features.

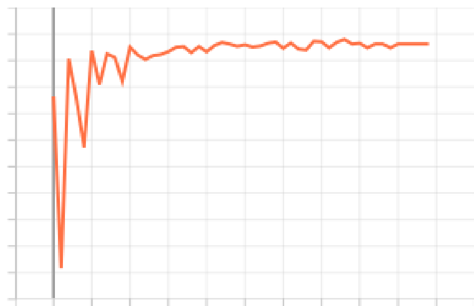



(b) Misclassified figurative paintings. Despite containing recognizable elements, the painter's style introduces abstraction, confusing the model.

Figure 4: Examples of misclassified paintings from the test set. Errors often stem from stylistic ambiguity or visual overlap between classes.



(a) Without scheduler.



(b) With scheduler.

Figure 5: Validation F1 score comparison with and without learning rate scheduler.

overall the model still shows good predictive ability. The results are presented in Table 4.

The confusion matrix on the test set (Figure 6) provides further insight into the model's behavior. Notably, the number of abstract paintings misclassified as figurative is very low (only 13), which confirms the model's strong performance in identifying abstract works. However, there is a relatively larger number of figurative paintings misclassified as abstract (201), which offsets the high recall for the abstract class and affects the overall F1-score.

#### 4.3.2 Experiment 2.1.1: Branch Importance

To better understand the contribution of each branch in the Two-Branch ResNet50 architecture, we conducted a series of experiments aimed at quantifying the importance of the full image branch versus the central patch branch in classification performance.

Our initial analysis focused on the final linear layer of the model, which directly connects the flattened feature vectors from each branch to the classification output. Since this layer does not introduce any interaction between the two branches, we can interpret the norm of the weights applied to each branch as a proxy for its relative contribution. As shown in Table 5, the norm of the weights connected to the patch branch is significantly lower than that of the full image branch. This suggests that the model inherently relies more on the global view of the painting than on the localized patch.

| | |
|---|---|
| **Model architecture** | Two-Branch ResNet50 |
| **Loss function** | CrossEntropyLoss (with class weights) |
| **Training epochs** | 50 |
| **Batch size** | 32 |
| **Learning rate** | 0.001 |
| **Data augmentation** | Enabled (standard) |
| **Scheduler** | Disabled |
| **Weighted loss** | Yes ($w_0 = 1.21$, $w_1 = 5.87$) |

Table 3: Configuration of the Two-Branch model (Experiment 2.1).

Table 4: Comparison of performance metrics across training, validation, and test sets (Experiment 2.1).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Accuracy | 0.9909 | 0.9235 | 0.8613 |
| F1-score | 0.9844 | 0.8631 | 0.8035 |
| Precision | – | 0.8577 | 0.7692 |
| Recall | – | 0.8689 | 0.8966 |

To validate this finding more concretely, we performed an ablation study by selectively deactivating one branch at a time during inference. We evaluated the full model, a version using only the full image branch, and one using only the patch branch. The results, shown in Table 6, reveal a striking gap in performance: while removing the patch branch has almost no effect on the model's accuracy or F1-score, removing the full image branch leads to a drastic performance collapse. The F1-score for the patch-only version drops to a mere 0.1139, confirming that the patch features alone are insufficient for reliable classification.

To further support these observations, we computed the mean norm of the feature vectors generated by each branch across the test set. As reported in Table 7, the norm of the patch branch outputs is nearly eight times smaller than that of the full image branch. This implies that not only are the patch features assigned smaller weights, but the features themselves carry less signal, contributing minimally to the final prediction.

**Discussion.** These findings raise an important question: why does the patch branch contribute so little to the model's decision-making? One possible explanation is related to the nature of the pre-trained ResNet50 backbone. These networks were originally trained on natural images (ImageNet), which are typically coherent full images containing complete objects. In contrast, our patch inputs are small, highly detailed crops that may not resemble the type of data the network is used to processing. Consequently, the ResNet may produce less informative or less discriminative features from these patches, explaining their lower norm and reduced impact.

Although the two-branch architecture was motivated by the idea of combining global and local information, our experiments suggest that, at least in this dataset and configuration, the local patch features contribute little to classification. This insight informs future architectural decisions: either the patch branch needs to be redesigned (e.g., via a separate backbone trained specifically on small regions), or additional mechanisms (such as attention) should be introduced to balance its contribution.

#### 4.3.3 Experiment 2.2: Gaussian Noise

To test the robustness of the model, we trained it on a modified version of the dataset where Gaussian noise with standard deviation 0.1 was added to all images. The training setup for this experiment is shown in Table 8.

Surprisingly, the model not only remained robust under noise, but also showed improved generalization on the test set, reaching an accuracy of 0.9151 and an F1-score of 0.8487. Both metrics

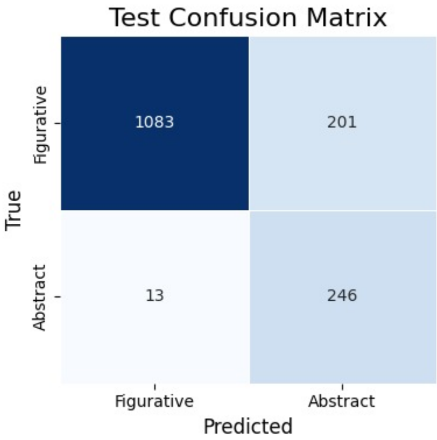



Figure 6: Test confusion matrix for Experiment 2.1.

Table 5: Norm of the linear weights per branch (Experiment 2.1).

| **Branch** | **Norm of weights** |
|---|---|
| Full image branch | [0.5778, 0.5850] |
| Patch branch | [0.4032, 0.4016] |

are higher than those obtained with clean training data, indicating that the added noise acted as an effective regularizer. The results can be seen in Table 9.

The confusion matrix for this experiment (Figure 7) shows a clear improvement in test performance. The number of misclassified samples is lower overall, and the distribution of errors is more balanced between both classes.

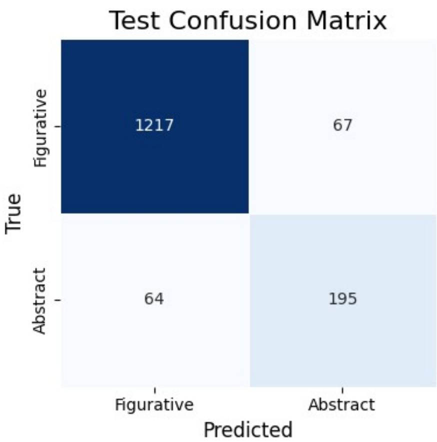



Figure 7: Test confusion matrix for Experiment 2.2 (with Gaussian noise).

This result suggests that the addition of Gaussian noise acted as a form of regularization, helping the model avoid overfitting and enhancing its ability to generalize. The Two-Branch ResNet50 proved not only expressive but also robust to moderate perturbations in the input data.

## 4.4 Grad-Cam

The results of the Grad-CAM analysis applied to the two-branched model—with a two-layer feed-forward network at the end—are presented in Figure 8. In this figure, the first row shows a complete figurative painting alongside the corresponding Grad-CAM heatmap, indicating where the model focuses its attention when predicting the ”figurative” class. As observed, the model tends to concentrate on human faces, which is consistent with the presence of identifiable objects

Table 6: Comparison of performance metrics across branch configurations (Experiment 2.1).

| Metric | Full Model | Only full image | Only patch |
|---|---|---|---|
| Accuracy | 0.9293 | 0.9300 | 0.8386 |
| F1-score | 0.7970 | 0.8007 | 0.1139 |
| Precision | 0.7698 | 0.7668 | 0.8472 |
| Recall | 0.8263 | 0.8378 | 0.8504 |

Table 7: Mean norm of feature vectors (Test dataset) (Experiment 2.1).

| Branch | Feature map mean norm |
|---|---|
| Full image branch | 4.894 |
| Patch branch | 0.613 |

in figurative art.

The second row presents an abstract painting and the regions the model considers relevant for classifying it as abstract. In contrast to the figurative case, the model appears to distribute its attention more broadly across the image, without focusing on any specific region. This may reflect the lack of concrete objects in abstract works.

In the third row, we examine the model's attention over the image patches. Interestingly, the Grad-CAM output shows that the model consistently focuses on the same area of the patch, regardless of the input image. This suggests that the model is not extracting meaningful information from the patch itself. The lack of variation in the model's attention across different patches implies that this branch contributes little to the final classification and may not be effectively leveraging the local image features.

### 4.4.1 Grad-CAM Conclusions

Based on these results, we can draw the following conclusions:

- The model's ability to classify an image relies primarily on its capacity to detect objects, faces, or other recognizable elements—likely due to its pre-training on ImageNet. If the model identifies a recognizable feature, it tends to classify the image as figurative. Conversely, if it cannot find any distinct or interpretable feature, it classifies the image as abstract.
- The model does not appear to learn from the style of the painting alone. If it did, we would expect it to attend to elements such as brush strokes, textures, or colour patterns, and its attention (in the patch) would vary depending on the image. This interpretation is supported by the fact that adding a second branch with patch-based processing did not lead to any performance improvement.
- This behavior is likely a consequence of how the model was trained. Since both branches and the final feedforward layers were fine-tuned jointly, the feedforward network may have prioritized the most informative input (the full image) while ignoring the less informative (patch-based branch). As a result, the gradients flowing to the patch branch during backpropagation were likely minimal, preventing it from learning meaningful representations.

## 5 Conclusions

The experiment proved to be a success. The model demonstrated a strong ability to distinguish between figurative and abstract paintings, achieving a macro F1-score (the average F1-score across classes) of nearly 90% in the best configuration, as shown in Table 2. This means that, on average,

| | |
|---|---|
| **Model architecture** | Two-Branch ResNet50 |
| **Loss function** | CrossEntropyLoss (with class weights) |
| **Training epochs** | 50 |
| **Batch size** | 32 |
| **Learning rate** | 0.001 |
| **Data augmentation** | Enabled (standard) |
| **Scheduler** | Disabled |
| **Weighted loss** | Yes ($w_0 = 1.21$, $w_1 = 5.87$) |
| **Noise std** | 0.1 |

Table 8: Configuration of the Two-Branch model with noise (Experiment 2.2).

Table 9: Comparison of performance metrics across training, validation, and test sets (Experiment 2.2).

| **Metric** | **Train** | **Validation** | **Test** |
|---|---|---|---|
| Accuracy | 0.9538 | 0.9190 | 0.9151 |
| F1-score | 0.9249 | 0.8607 | 0.8487 |
| Precision | – | 0.8433 | 0.8472 |
| Recall | – | 0.8819 | 0.8504 |

the model achieved 90% F1-score for each class, an outstanding result, especially considering that it was fine-tuned on the painting dataset for only a few epochs.

However, this exceptional performance is not due to the model learning the artistic style of the painters. Rather, it stems from the model's ability to identify recognizable elements in figurative paintings, such as faces, trees, or boats. When such elements are present, the model classifies the image as figurative. When it fails to detect anything familiar, it classifies the image as abstract.

When comparing the three different models, we found some interesting results. First, there was no significant difference between the two-branch ResNet model and the same architecture with an added two-layer feedforward network at the end. All performance metrics and predictions were essentially identical. Therefore, we will not analyze that variant further and will focus the comparison on the fine-tuned ResNet-50 and the two-branch ResNet.

In terms of accuracy, F1-score, and computational efficiency, the fine-tuned ResNet-50 outperformed the two-branch architecture, reaching an F1-score close to 90%, while the latter failed to surpass 85%. A possible explanation for this is that, given the way the models were trained, the ResNet processing the patch input did not contribute sufficient information for the projection layer to consider it relevant during classification. As a result, the patch-branch weights received weak gradients, limiting their learning. As a result, the patch input only contributed to a shift in the final feature distribution, without adding meaningful information for classification, which is evident in the confusion matrix (see Figure 6), where we observe a tendency to misclassify figurative images as abstract. Despite this, the model was able to accurately classify a higher number of abstract images.

# 6 Future Work

Although the results obtained in this project are promising, there are several directions that could be explored to improve performance and deepen our understanding of the classification task.

**Improving Patch Selection.** In our two-branch architecture, the patch branch failed to contribute meaningful information to the final classification. One hypothesis is that the fixed central crop may not capture relevant visual elements. A potential improvement is to apply edge detection (e.g., using the Sobel operator or Canny filter) to guide patch selection toward more informative regions—such as areas with high texture variation or structural complexity—rather than relying on a fixed location.

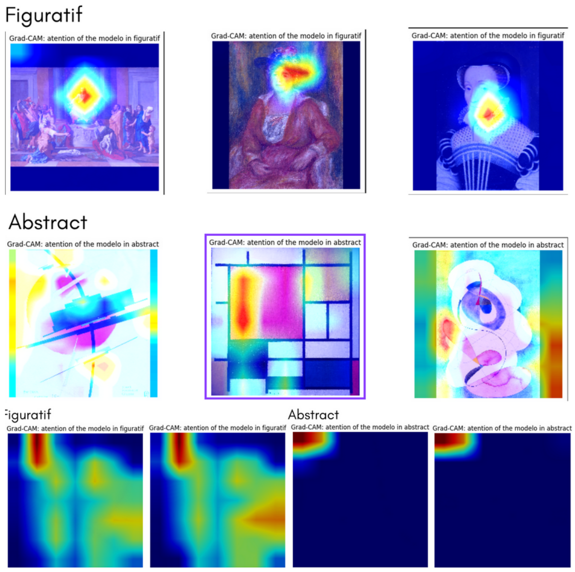



Figure 8: Figure: Grad-CAM visualizations from the two-branch ResNet model with a two-layer feed forward at the end. Top: Attention on whole figurative images, the model focuses on recognizable elements like faces. Middle: Attention on whole abstract images, focus is diffuse and less specific. Bottom: Attention from the patch branch, similar focus across inputs, suggesting low contribution to classification.

**Data Augmentation with GANs.** The strong class imbalance between figurative and abstract paintings remains a challenge. While we addressed this using a weighted loss, another promising strategy is to synthetically augment the minority class (abstract paintings) using Generative Adversarial Networks (GANs). A well-trained GAN could produce realistic abstract artworks that enrich the training set, improve model generalization, and reduce overfitting toward the dominant class.

**Label Quality and Semi-Supervised Correction.** As noted during our qualitative error analysis, some paintings may be mislabelled—especially those near the abstract/figurative boundary. Improving label quality through manual review would enhance training reliability. Alternatively, semi-supervised learning methods could be explored to identify and correct potential labeling errors automatically, using model uncertainty or consistency-based approaches.

Overall, these extensions aim to make the model not only more accurate but also more robust and interpretable in its treatment of complex visual and artistic data.

**Use Contrastive Learning to Improve the Embedding Space.** Another way to improve the model's behavior is to structure the embedding space such that vectors from the same class are closer together, and those from different classes are further apart. This could be achieved using contrastive learning, as proposed in the article [5]. Such a loss function would encourage better separation in the embedding space and allow the use of a distance metric to assess how abstract or figurative a painting is, potentially improving both model performance and interpretability.

**Train the Whole Model and the Patch Model Separately** One possible reason the patch model failed to learn meaningful features is that it was trained jointly with the whole-image branch, which contains much richer information for classification. To address this, a potential improvement would be to train the two ResNet branches independently and later combine them using a feedforward network as a classifier. This setup could encourage both branches to learn complementary features and improve overall performance.

## References


[1] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[2] Adrian Lecoutre, Benjamin Negrevergne, and Florian Yger. "Recognizing Art Style Automatically in Painting with Deep Learning". In: *Proceedings of the Ninth Asian Conference on Machine Learning*. Ed. by Min-Ling Zhang and Yung-Kyun Noh. Vol. 77. Proceedings of Machine Learning Research. Yonsei University, Seoul, Republic of Korea: PMLR, 15–17 Nov 2017, pp. 327–342. URL: https://proceedings.mlr.press/v77/lecoutre17a.html.

[3] Raffaele Gaetano et al. "MRFusion: A Deep Learning architecture to fuse PAN and MS imagery for land cover mapping". In: *CoRR* abs/1806.11452 (2018). arXiv: 1806.11452. URL: http://arxiv.org/abs/1806.11452.

[4] Ramprasaath R. Selvaraju et al. "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization". In: *CoRR* abs/1610.02391 (2016). arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

[5] Amit Mandelbaum and Daphna Weinshall. "Distance-based Confidence Score for Neural Network Classifiers". In: *CoRR* abs/1709.09844 (2017). arXiv: 1709.09844. URL: http://arxiv.org/abs/1709.09844.