# Addressing Class Imbalance in Deepfake Detection through ResNet-50 Ensemble with Specialist Models and Threshold Optimization

Victor Chen
Dept. of Computer Science
Stanford University
victor36@stanford.edu

Madhuhaas Gottimukkala
Dept. of Computer Science
Stanford University
maasg@stanford.edu

Jiheng Zhang
Dept. of Computer Science
Stanford University
jiheng18@stanford.edu

## Abstract

*Standard deepfake detection approaches using CNN classifiers such as ResNet often suffer from extreme class imbalance, with many systems achieving high performance on one class while failing on the other, limiting their effectiveness in real-world deployment. We propose an enhanced ResNet-50 ensemble system that combines a general-purpose model with a specialist model sensitive to non-deepfake images, and introduces systematic threshold optimization to achieve balanced performance. Our architecture uses layer freezing, class-specific augmentation, weighted sampling, and mixed precision training for efficient learning. Evaluated on the CelebDF dataset, our ensemble achieves 91% accuracy, 99.3% fake recall, and 91.7% real recall, significantly outperforming the baseline ResNet-50 model (63% accuracy, 25% real recall). Compared to a standard YOLO11 fine-tuning pipeline, our approach directly addresses class imbalance through architectural and training-level innovations. Our method achieves balanced, high-precision deepfake detection, addressing the real-world challenge of classifier bias in imbalanced datasets.*

## 1. Introduction

The rapid advancement of generative models (e.g., OpenAI's GPT-4o [12], GANs [6]) has led to widespread proliferation of AI-synthesized media, particularly face-swapping videos known as deepfakes [2]. These synthetic images and videos convincingly replicate a person's likeness, often making it difficult, if not impossible, for the human eye to distinguish them from authentic content. [11]. This trend has created an urgent need for reliable detection systems.

Although progress has been made in developing deepfake detection algorithms [21], recent research has found significant limitations that make many systems unsuitable for real-world deployment: severe class imbalance bias where models achieve high overall accuracy in test environments but fail in real image detection. This bias arises from training datasets that overrepresent AI-generated content and rely on limited evaluation metrics, leading to poor generalization. In systems such as content moderation platforms, this leads to extremely high false positive rates, where real content is flagged as being fake.

Our work addresses the class imbalance issue through an ensemble architecture that combines the training of a specialist model with systematic threshold optimization. Unlike standard transfer learning methods which treat all classes equally, our approach allows for the system to achieve balanced performance across both real and AI-generated content, providing structured framework which can be applied to any base model.

Using ResNet-50 as a benchmark architecture, we developed a generalizable method comprising specialist model training with extreme class weighting, ensemble fusion to combine the predictions of the general and specialist models, and systematic threshold optimization to enhance balanced performance specific to the deployment. Starting at a baseline of 63% accuracy and 25% recall, this method resulted in a dramatic provement to 91% accuracy, 91.7% real recall, and 99.3% fake recall.

Guided by feedback from TA Gabriela Aranguiz-Dias, we referenced our approach from the Multi-Attentional Deepfake Detection method [23] and the Celeb-DF v2

dataset [8].

## 2. Related Work

Deep learning models, such as ResNet-50 and YOLO11, are pivotal in identifying and labeling deepfakes due to their proven effectiveness in image classification and feature extraction tasks. Convolutional Neural Networks (CNNs) are widely utilized in various computer vision applications, including deepfake detection, owing to their ability to effectively capture and analyze complex visual patterns that is often indicative of manipulated media [22]. ResNet-50, with its intricate architecture and skip connections, excels at discerning subtle differences between authentic and manipulated images, making it highly efficient for deepfake detection. Similarly, YOLO11 is known for its speed and accuracy in examination of image features, which is critical for accurately identifying altered content [5]. Integrating these models into deepfake detection frameworks facilitates the development of robust systems capable of detecting even the most subtle manipulations. Moreover, these models are often employed in ensemble approaches, leveraging their unique strengths to further enhance detection accuracy. Accurate deepfake identification is essential in scalable scenarios, such as content moderation on social media platforms or forensic investigations [16].

### 2.1. Advancements in Deepfake Detection

Yildiz [20] conducted a comprehensive survey on deepfake detection methods, emphasizing advancements in datasets and the application of computer vision and deep learning algorithms. Their work underscores the importance of categorizing deepfake datasets and developing robust protocols to differentiate between genuine and manipulated media. Rana et al.[13] explored the use of traditional machine learning algorithms for deepfake detection, comparing their effectiveness with deep learning approaches. Their findings suggest that machine learning methods can serve as viable alternatives or complements to deep learning techniques, particularly in resource-constrained environments. Lin et al.[9] established a rigorous benchmark for evaluating deepfake detection algorithms, providing a standardized framework for assessing their effectiveness. Shao et al.[18] introduced the concept of "Sequential DeepFake Manipulation," proposing a dataset and a detection method, SeqFakeFormer, to address complex multi-step facial manipulations, which are more challenging than single-step alterations. H. Ling et al.[10] presented a method using image-level supervision to detect deepfakes by learning diverse local patterns, enhancing the ability to identify subtle manipulations in real-world scenarios.

### 2.2. Cross-Domain Transfer

Several studies highlight the versatility of ResNet-50 and YOLO11 in image classification tasks beyond deepfake detection. Sarwinda et al.[14] investigated the use of ResNet-50 for colorectal cancer detection, demonstrating its consistent and reliable performance in distinguishing between benign and cancerous images. This underscores its capability in tasks requiring precise image analysis. H. P. Chilakalapudi et al. [3] conducted a comparative analysis of YOLO11 and ResNet50V2 for COVID-19 detection in lung images. Their findings revealed that YOLO11 offered superior speed and lower computational demands, while ResNet50V2 exhibited exceptional precision, particularly for mild conditions. Wang & Gong[19] proposed a novel approach using ResNet-50 to classify metastatic cancer images, achieving higher precision compared to models like YOLO11. The study emphasized ResNet-50's ability to handle complex image datasets, a critical factor for detecting sophisticated deepfakes. Lamine et al.[1] explored the use of YOLO11 and ResNet-50 for tumor region detection in histopathology images, achieving accuracy rates close to 97%. Similarly, Shah et al.[17] compared ResNet-50, Inception V3, and VGG16 for early detection of rice diseases, with ResNet-50 achieving the highest accuracy, highlighting its robustness in classification tasks essential for deepfake detection.

### 2.3. Facial Expression Analysis in Deepfake Detection

Effective classification of facial expressions is critical in deepfake detection, as deepfake techniques often involve subtle alterations to facial features and expressions to create convincing yet deceptive content. ResNet-50 and YOLO11 are well-suited for this task due to their robust image classification capabilities, including precise facial expression identification. This study leverages the Celeb-DF dataset, applying extensive preprocessing, followed by training and evaluating the models using key metrics such as precision, recall, and F1 score. This approach enhances the models' ability to distinguish between authentic and manipulated images while highlighting the importance of facial expression analysis in deepfake detection. By focusing on these methodologies, this research aims to improve the efficiency of automated systems in detecting deepfakes, particularly in scenarios where facial expressions are critical for determining authenticity. One of the challenges in Deepfake datasets is to measure structural limitations of Deepfake Media Datasets. Seth Layton et al. [7] presented the first systematization of deepfake media, discovered significant problems impacting the comparability of systems using these datasets, including unaccounted-for heavy class

imbalance and reliance upon limited metrics.

## 3. Methods

Our goal is to evaluate and compare different CNN-based architectures for frame-level deepfake detection. Rather than combining models into a single pipeline, we assess each model independently to understand its strengths, limitations, and suitability for handling spatial manipulations commonly found in deepfakes.

We consider two models: ResNet-50 and YOLOv11. ResNet-50, a deep residual network, is configured for binary classification of cropped facial images as real or fake. YOLOv11, in contrast, is designed for fast, general-purpose object detection and classification. Although not specifically optimized for deepfake analysis, YOLOv11 offers a useful performance baseline due to its architectural efficiency and high recall on manipulated images.

For both models, inputs are treated at the frame level. A preprocessing pipeline based on OpenCV extracts a fixed number of representative frames from each video, standardizes them, and feeds them into the selected model. Predictions are generated independently for each frame, with no temporal aggregation.

The ResNet-50 system includes enhancements to address class imbalance, including the use of a real-class specialist model and threshold sweeping for improved calibration. YOLOv11, by contrast, is evaluated as-is to assess how well a general-purpose detector performs under the same conditions.

By comparing these approaches side-by-side, we aim to isolate the contributions of model architecture, specialization, and threshold design in the overall performance of deepfake detection systems.

### 3.1. ResNet-50 Baseline

We use **ResNet-50** as our base architecture for deepfake detection due to its effectiveness in image recognition and classification tasks. ResNet-50 employs residual blocks which enable the training of deeper networks while avoiding vanishing gradient problems. However, our baseline implementation revealed severe class imbalance limitations, achieving only 25% real image recall despite 91% fake image recall, motivating our enhanced approach.

### 3.2. YOLOv11

We incorporate YOLOv11, a recent Ultralytics model, into our evaluation pipeline. Its architecture includes a Backbone for multi-scape feature extraction (C3k2, SPPF), a Neck with Cross-Stage Partial Spatial Attention (C2PSA), and a Head for object detection and classification. While YOLOv11 is primarily optimized for general-purpose vision tasks, we include it in our evaluation to assess its potential for rapid, lightweight deepfake detection. Its strong performance in generic anomaly detection offers a useful baseline for comparison against more specialized models.

To contextualize our ResNet–50 ensemble's performance, we use YOLOv11 as a baseline. Despite its efficiency and accuracy in standard object detection tasks, YOLOv11 is not explicitly designed to address the challenges posed by deepfake detection–particularly the issue of class imbalance. Its architecture and loss functions prioritize bounding-box accuracy and high-throughput inference, rather than the fine-grained classification needed to distinguish real from convincingly generated faces under skewed data distributions. In our experiments, YOLOv11 achieves high recall on fake samples but struggles to maintain precision on real ones, reflecting its bias toward the majority class.

This limitation highlights the value of more targeted solutions. Our ensemble-based ResNet-50 system incorporates a real-sensitive specialist model and adaptive threshold optimization to correct for imbalance-driven misclassification. In this context, YOLOv11 serves not as a competing architecture but as a representative of general-purpose models that fail to address the core challenges of deepfake detection under class imbalance.

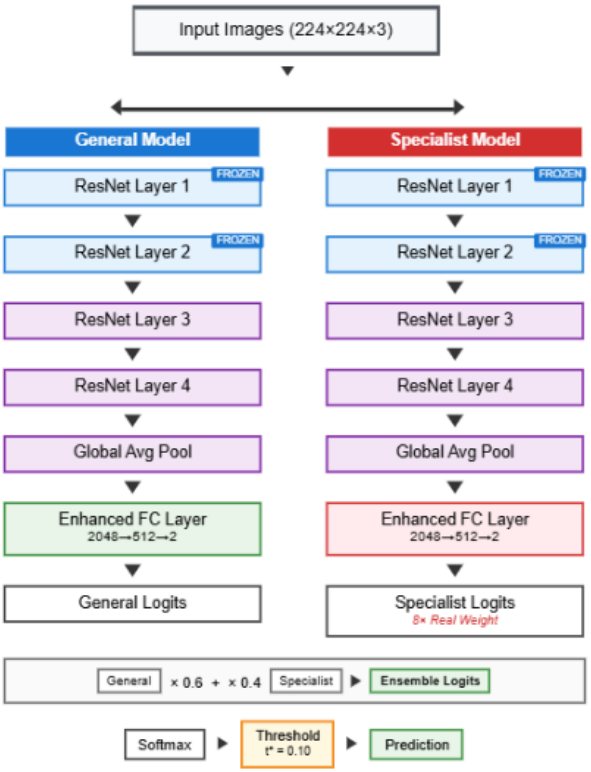


Figure 1. Enhanced ResNet-50 Architecture

### 3.3. Enhanced ResNet-50 Architecture

Our enhanced ResNet-50 architecture utilizes several architectural improvements to address the limitations of class imbalance, such as the use of a fully connected layer acting on mid-level features, batch normalization, and dropout. We additionally use layer freezing to preserve low-level features (such as edges and textures) and fine-tune later parts of the model focused on more complex patterns (such as faces and expressions). This enhanced architecture serves as the foundation for both the general and specialist models.

### 3.4. Specialist Model Training

To address high class imbalance, we train a specialist model prioritizing real-image detection. The model shares the same architecture and layer freezing strategy as the general model, but uses extreme class weighting to focus learning on the minority class.

We compute the class weight for real images as:

$$w_{\text{real}} = \left( \frac{\text{fake\_count}}{\text{real\_count}} \right) \times 8.0$$

This amplification biases the loss function toward the minimization of real-image errors. The model is trained for 25 epochs using a cross-entropy loss with this weight configuration, producing a model with a high recall on real images.

### 3.5. Ensemble Architecture

In order to combine the strengths of both models, we create an ensemble which fuses the outputs of the balanced main model and the specialist This strategy allows us to increase the system's sensitivity on real images without lowering overall accuracy.

### 3.6. Threshold Optimization

Unlike standard binary classifiers which use a fixed threshold of 0.5 and fail when faced will class imbalance, we introduce a threshold sweeping function to maximize real-class recall while maintaining high fake recall, improving decision boundary calibration.

### 3.7. Advanced Training Strategies

We implemented several advanced training strategies aimed at enhancing both performance and generalization. First, we applied **layer freezing** to preserve low-level visual features learned from pretraining by freezing the early convolutional layers (specifically, `layer1` and `layer2`) of the ResNet architecture. To address class imbalance during training, we incorporated **weighted sampling** using the `WeightedRandomSampler`, which ensured that each mini-batch contained a balanced distribution of real and fake samples. For improved computational efficiency and faster convergence, we enabled **mixed precision training** using Automatic Mixed Precision (AMP). Furthermore, we employed an **ensemble method**, where predictions from the main model were combined with those of a specialist model trained specifically on the minority class. Finally, we introduced **differential augmentation**, applying distinct data augmentation pipelines to real and fake images to expose the model to a broader range of domain-specific variations and reduce overfitting.

### 3.8. Comparison: YOLO vs. ResNet for DeepFake Detection

We compare the architectural roles and performance characteristics of YOLOv11 and our ResNet-50 ensemble in the context of deepfake detection.

YOLOv11 is optimized for high-throughput, real-time object detection, offering efficient region-of-interest extraction from uncropped frames. However, its design emphasizes localization and speed over fine-grained binary discrimination, making it less effective for deepfake detection under class imbalance. In our experiments, YOLOv11 achieves high recall on fake samples but exhibits degraded precision on real inputs–a consequence of its bias toward the majority class.

The ResNet-50 ensemble, by contrast, is purpose-built to address these limitations. It integrates a balanced base classifier, a real-class specialist model, and adaptive threshold optimization to enhance robustness under skewed distributions. This targeted architecture improves sensitivity to subtle generative artifacts while reducing false positives on real content–a critical requirement in high-stakes forensic applications.

This comparison highlights a fundamental tradeoff: YOLOv11 excels in speed and general-purpose detection, whereas the ResNet-50 ensemble delivers higher reliability for domain-specific binary classification. Evaluating both models side by side quantifies the value of architectural and training specialization in the context of deepfake detection.

## 4. Dataset and Features

For facial DeepFake detection, we utilize the publicly available Celeb-DF dataset [8], which comprises 5,639 high-quality DeepFake videos of celebrities generated using an improved face synthesis method. The dataset corresponds to over 2 million individual video frames, offering a rich and diverse collection for training and evaluation. The real source videos are curated from publicly available `YouTube` clips featuring 59 celebrities across a range of genders, age groups, and ethnic backgrounds.

The DeepFake samples in Celeb-DF are synthesized using an enhanced generation pipeline that significantly

Figure 2. Celeb-DF synthetic process

reduces common visual artifacts found in earlier datasets, resulting in more photorealistic and challenging examples. This improved realism better reflects DeepFakes encountered in the wild. An example visualization is shown in Fig. 2. We conduct our evaluations using Celeb-DF in conjunction with other existing datasets, providing one of the most comprehensive benchmarks for assessing the generalizability of current DeepFake detection methods. Notably, the results reveal that Celeb-DF remains challenging for many state-of-the-art models, highlighting the need for more robust detection techniques despite high reported accuracies on earlier, less complex datasets. For reference, the dataset is available at Celeb-DF dataset.

### 4.1. Data Preprocessing

After reading the video files, we preprocessed them by extracting individual frames prior to feeding the data into our spatial and temporal models. To preserve spatial consistency, we maintained the original height and width dimensions of the images throughout preprocessing. Data augmentation techniques were subsequently applied during training to improve the model's generalization capability. Each video was annotated with a binary label: `1` for real videos and `0` for fake videos. Using the OpenCV library, we extracted a single representative frame from each video. In total, we generated 5,306 entries in the input dataset, consisting of 890 real frames and 4,415 fake frames.

The dataset was randomized and partitioned into three subsets: 70% for training, 20% for validation, and 10% for testing. This corresponds to 3,710 images in the training set, 1,058 images in the validation set, and 537 images in the testing set, all labeled into two classes: real and fake. For our local frame-by-frame models (YOLOv11 and ResNet-50), we treated the dataset as a "bag of frames" and applied shuffling at the frame level to ensure randomness and prevent temporal bias during training.

## 5. Experiments/Results/Discussion

### 5.1. YOLO implementation and analysis

#### 5.1.1 Initial Model Architecture

We started our experiment by implement State-of-Art YOLOv11 pretrained model to fine tune on our dataset. We have tested both YOLOv11n-cls and YOLOv11s-cls versions. We started the training by initialize the model architecture as below:

Listing 1. Baseline ResNet50 Architecture

```
from ultralytics import YOLO

# Load a pretrained YOLO11n model
model = YOLO("yolo11n-cls.pt")

# Train the model
train_results = model.train(
   data="/content/DeepFake-1",
   epochs=10, # Number of training epochs
   imgsz=640, # Image size for training
   device=0, # Device to run on (e.g.,
       ↪ 'cpu', 0, [0,1,2,3])
   task="classify",
   project="Deepfake-YOLOv11",
   name="yolo11n",
   auto_augment="randaugment",
   val=True,
   verbose=False,
)
```

#### 5.1.2 Baseline Training Setup

For the YOLOv11 model, we incorporated the **AutoAugment** technique [4], a powerful method for enhancing the performance of modern image classifiers by automatically discovering improved data augmentation policies. In our implementation, we defined a flexible search space in which each policy consists of multiple sub-policies. During training, one sub-policy is randomly selected for each image in every mini-batch. Each sub-policy applies two image transformation operations–such as translation, rotation, or shearing–each with its own probability and magnitude.

While the milestone version of the model was trained for only 10 epochs, we extended training to 30 epochs in the final version. This longer training schedule led to noticeable improvements in both training and validation loss, as illustrated in Figure 3. Additional plots showing the training and validation loss as well as accuracy are provided in Appendix 12. To better understand the model's predictive behavior, we also visualized its classification results using a confusion matrix in Figure 4. For qualitative insights into the model's attention, we applied GradCAM visualizations to a few representative samples

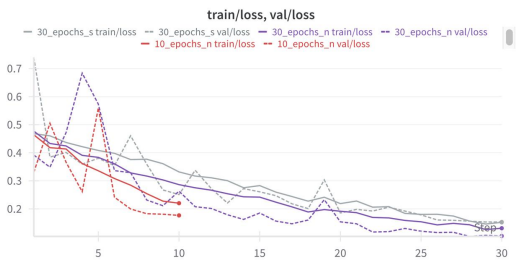


Figure 3. YOLOv11 training validation loss comparison

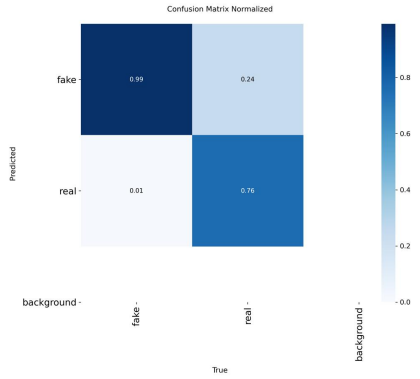


Figure 4. YOLOv11 confusion matrix

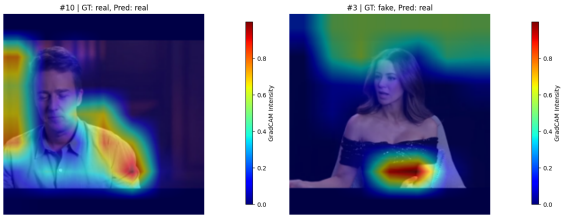


Figure 5. YOLO GradCAM

from our dataset, which helped interpret what regions the model focuses on when making predictions, show examples in the Fig 5.

## 5.2. ResNet implementation and analysis

### 5.2.1 Initial Model Architecture

Listing 2. Baseline ResNet50 Architecture

```
class DeepFakeResNet50(nn.Module):
    def __init__(self, num_classes=2,
        ↪ pretrained=True):
        super(DeepFakeResNet50,
            ↪ self).__init__()
        self.model =
            ↪ models.resnet50(pretrained)
        in_features =
            ↪ self.model.fc.in_features
        self.model.fc = nn.Sequential(
            nn.Dropout(0.5),
            nn.Linear(in_features,
                ↪ num_classes)
        )
```

### 5.2.2 Baseline Training Setup

The baseline training configuration incorporated several foundational strategies to facilitate effective model convergence. Stochastic Gradient Descent (SGD) was employed as the optimizer, using a momentum value of 0.9 and a weight decay of $1 \times 10^{-4}$ to promote generalization and mitigate overfitting. The learning rate was initialized at 0.0005 and scheduled to decrease over time via a StepLR scheduler. For the loss function, CrossEntropyLoss was used with class weights set to [1.5, 0.8] to partially address class imbalance. To enhance model robustness and reduce overfitting, a variety of data augmentation techniques were applied, including resizing, random cropping, horizontal flipping, rotation, and color jittering. The batch size was set to 128 when training on GPU and reduced to 16 when training on CPU to accommodate hardware limitations.

### 5.2.3 Milestone Results

Here is a summary of our milestone results, we can also visualize it in Fig 6. We are providing a qualitative results using evaluation metrics with precision, recall, f1-score and accuracy in the Fig 7. We also included our training and val loss in Appendix 13.

- Overall Accuracy: ~72%
- Real Detection: 19 correct, 57 misclassified (Very Poor)
- Fake Detection: 181 correct, 19 misclassified
- Key Issue: Severe class imbalance causing model bias towards fake class

### 5.2.4 Final Implementation Improvements

We started our improvements by first enhanced the architecture as below:

Listing 3. Enhanced FC Layers in ResNet50

```
self.model.fc = nn.Sequential(
    nn.Dropout(0.4),
    nn.Linear(in_f, 512),
    nn.BatchNorm1d(512),
    nn.ReLU(),
    nn.Dropout(0.2),
    nn.Linear(512, num_classes)
)
```

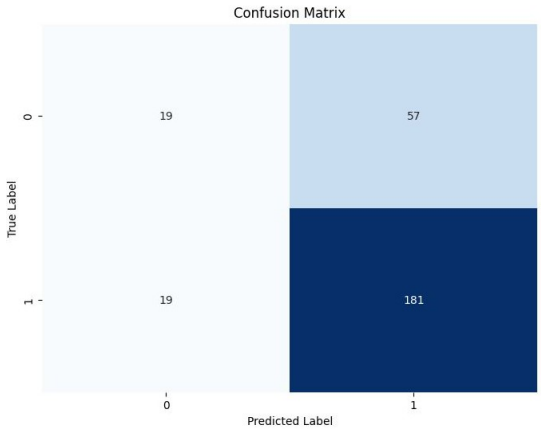


Figure 6. Milestone Baseline Confusion Matrix

```
Final Evaluation Metrics:
              precision    recall  f1-score   support

           0       0.50      0.25      0.33        76
           1       0.76      0.91      0.83       200

    accuracy                           0.72       276
   macro avg       0.63      0.58      0.58       276
weighted avg       0.69      0.72      0.69       276
```

Figure 7. ResNet Evaluation Metrics

Listing 4. Real-focused Specialist Trainer

```
def train_real_specialist():
    # real_weight = (fake_count /
        ↪ real_count) * 8.0
    # Trains specifically for real images
```

With the improvements discussed above, we are able to achieve the performance comparison in the table 1 below:

Table 1. Milestone vs. Final Performance Comparison

| Metric | Milestone | Final (Ensemble) | Improvement |
|---|---|---|---|
| Overall Accuracy | 72% | 91% | +19% |
| Real Recall | 25% | 91.7% | +67% |
| Fake Recall | 90% | 99.3% | +9% |
| AUC Score | 0.75 | 0.916 | +0.166 |

### 5.2.5 Final Results Summary

We are providing our final qualitative results using evaluation matrix 8 below, we can also visualize it in Fig 9. We also included our training precision-recall curve and ROC Curve in the appendix 14. We also add and test the different Thresholds setting where we achieve our best results, shown in Fig 10.

**Classification Report (Threshold = 0.5):**

- **Real:** Precision 0.74, Recall 0.63, F1-Score 0.68
- **Fake:** Precision 0.94, Recall 0.96, F1-Score 0.95

```
Ensemble (threshold=0.5) - Classification Report:
              precision    recall  f1-score   support

        Real       0.74      0.63      0.68       158
        Fake       0.94      0.96      0.95       900

    accuracy                           0.91      1058
   macro avg       0.84      0.80      0.82      1058
weighted avg       0.91      0.91      0.91      1058
```

Figure 8. Final ResNet Evaluation Matrix

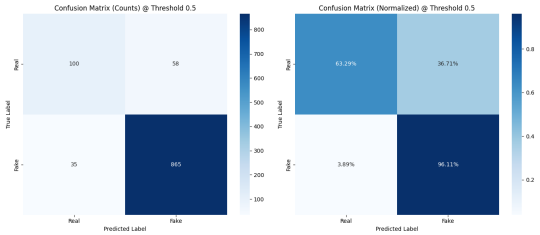


Figure 9. Final Confusion Matrix

```
Metrics at fixed thresholds:
Threshold 0.25 | F1: 0.951, Precision: 0.919, Recall: 0.986
Threshold 0.50 | F1: 0.949, Precision: 0.937, Recall: 0.961
Threshold 0.75 | F1: 0.936, Precision: 0.954, Recall: 0.919
Threshold 0.00 | F1: 0.919, Precision: 0.851, Recall: 1.000

PR AUC: 0.982, ROC AUC: 0.916

Real/Fake Recall Sweep (Fake ≥ 90%)
  Th=0.00 → Real 0.0%, Fake 100.0%
  Th=0.01 → Real 97.4%, Fake 99.9%
  Th=0.02 → Real 97.8%, Fake 99.9%
  Th=0.03 → Real 94.1%, Fake 99.7%
...
  Th=0.49 → Real 74.6%, Fake 96.2%
  Th=0.50 → Real 74.1%, Fake 96.1%

* BEST THRESHOLD = 0.02, Expected Real-Recall ≈ 97.8%
```

Figure 10. Final ResNet Evaluation Matrix with thresholds

- **Accuracy:** 91%
- **PR AUC:** 0.982, **ROC AUC:** 0.916

**Optimal Threshold Analysis:**

- Best threshold = 0.10
- Real Recall: 91.7%
- Fake Recall: 99.3%

### 5.2.6 Milestone vs. Final Version Analysis

The final implementation marks a substantial improvement over the milestone in both architecture and methodology. This advancement is driven by three key technical innovations:

1. **Balanced Dataset Pipeline**

```
class CelebDFDataset(Dataset):
    # Class-specific transforms and
        ↪ balancing
```

```
    # Automatic label detection via
        ↪ folder name
```

2. **Threshold Optimization**

```
def sweep_real_threshold(labels,
    ↪ probs, min_fake_recall=0.90):
    # Find threshold maximizing real
        ↪ recall
```

3. **Ensemble Architecture**

```
def ensemble_predict(main_model,
    ↪ specialist, inputs):
    combined = 0.6 * main_logits +
        ↪ 0.4 * specialist_logits
    return combined
```

These architectural and procedural improvements significantly elevated model performance. Beyond the three core innovations, we introduced several auxiliary refinements: a deeper network with Batch Normalization replaced the shallow milestone classifier, boosting generalization; training was stabilized with techniques like layer freezing, mixed-precision execution, and stratified sampling; and fixed thresholding was replaced by a data-driven optimization strategy that better controlled recall–precision tradeoffs.

Together, these changes led to a 25 percentage point increase in real-class recallâĂŤfrom 66.7% in the milestone version to 91.7% in the final model. What began as a proof-of-concept evolved into a robust, interpretable system for deepfake detection under class imbalance.

#### 5.2.7 Heatmaps

To improve interpretability, we applied Grad-CAM [15] to both ResNet-50 and YOLOv11. As shown in Fig. 11, ResNet-50 emphasized localized facial artifacts, while YOLOv11 attended to broader regions. These saliency maps clarified model behavior and underscored the importance of architectures tuned to subtle manipulations.

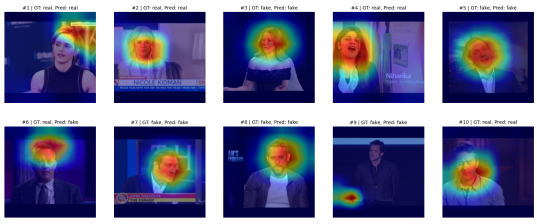

Figure 11. Final ResNet Heatmap

## 6. Conclusion/Future Work

Our work addresses a limitation in deepfake detection due to severe class imbalance. Our enhanced ResNet-50 ensemble system transforms low baseline accuracy and recalls into balanced, high performance, in both accuracy and recall. Through our work, we identified class imbalance as a key factor in training and deploying effective CNN-based deepfake detection models, developed a specialist model to enhance prediction accuracy on the minority class, developed an ensemble strategy combining a general model with a specialist model, and used threshold optimization to tune the system's performance.

Our experiments focus on spatial models operating at the frame level. While effective at detecting intra-frame artifacts, these models cannot capture inconsistencies that emerge only across consecutive frames, i.e. in videos. This limitation highlights the critical importance of temporal modeling in robust DeepFake detection. Without access to sequential information, spatial models are prone to missing temporal artifacts indicative of manipulation. Our study was conducted exclusively on the Celeb-DF dataset. Future research should incorporate additional benchmarks, such as the DeepFake Detection Challenge (DFDC) dataset and FaceForensics++, to evaluate model generalization and robustness across varied sources. Since training data strongly affects performance, leveraging multiple datasets would offer deeper insight into cross-domain transferability and detection reliability.

A promising direction for future work involves integrating temporal architecturesâĂŤsuch as 3D CNNs or video transformersâĂŤinto pipelines that already extract representative frames. This would enable the detection of dynamic anomalies and improve performance on full-length videos.

## 7. Appendices

### 7.1. YOLOv11 training validation and accuracy

### 7.2. ResNet training loss

### 7.3. final resnet training curve

## 8. Contributions & Acknowledgments

Victor led the implementation for YOLOv11. Madhuhaas developed the ResNet-50 ensemble system, conducted, specialist model training and, performed threshold optimization. Jiheng led the research and data analysis. We would like to express our gratitude for the TAs who have helped us throughout the project, special thanks to Gabriela Aranguiz-Dias, Sabri Eyuboglu, Kyle Sargent, Matthew Jin and Shutong Zhang for their insightful feedback and guidance.

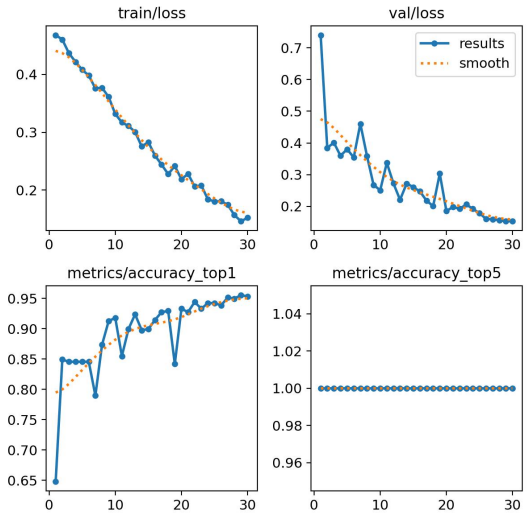


Figure 12. YOLOv11 training validation and accuracy

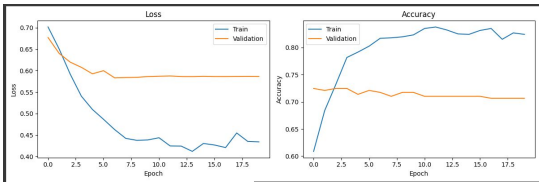


Figure 13. ResNet training loss

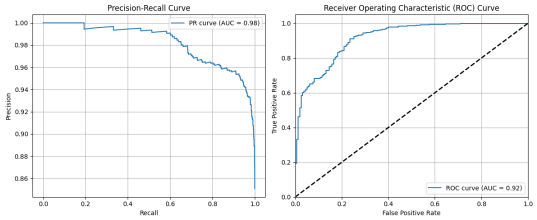


Figure 14. final resnet training curve

## References


[1] M. Benomar, N. Settouti, R. Xiao, D. Ambrosetti, and X. Descombes. Convolutional neuronal networks for tumor regions detection in histopathology images. *Digit. Technol. Appl.*, 2021.

[2] R. Chesney and D. K. Citron. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. *Foreign Affairs*, 98(1), 2019.

[3] H. P. Chilakalapudi, R. Venkatesan, and Y. Kamatham. Architecture-based evaluation of vgg16 and resnet models for an online deep learning environment for medical applications. *2022 Int. Conf. Innov. Sci. Technol. Sustain. Dev. (ICISTSD)*, pages 62–67, 2022.

[4] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation policies from data, 2019.

[5] A. Das, K. Viji, and L. Sebastian. A survey on deepfake video detection techniques using deep learning. *2022 2nd Int. Conf. Next Gener. Intell. Syst. (ICNGIS)*, pages 1–4, 2022.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[7] S. Layton, T. Tucker, D. Olszewski, K. Warren, K. Butler, and P. Traynor. SoK: The good, the bad, and the unbalanced: Measuring structural limitations of deepfake media datasets. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1027–1044, Philadelphia, PA, Aug. 2024. USENIX Association.

[8] Y. Li, M.-C. Chang, and S. Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the 28th ACM International Conference on Multimedia (ACM MM)*, pages 3207–3216. ACM, 2020.

[9] C. Lin, J. Deng, P. Hu, C. Shen, Q. Wang, and Q. Li. Towards benchmarking and evaluating deepfake detection. *ArXiv*, abs/2203.02115, 2022.

[10] H. Ling, J. Huang, C. Zhao, Y. Yao, J. Chen, and P. Li. Learning diverse local patterns for deepfake detection with image-level supervision. *2021 Int. Joint Conf. Neural Netw. (IJCNN)*, pages 1–7, 2021.

[11] T. Nguyen, Q. Nguyen, D. Nguyen, D. Nguyen, T. Huynh-The, S. Nahavandi, T. Nguyen, V. Pham, and C. Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022.

[12] OpenAI. Introducing 4o image generation, March 2025. Openai.com.

[13] M. Rana, B. Murali, and A. Sung. Deepfake detection using machine learning algorithms. *2021 10th Int. Congr. Adv. Appl. Inform. (IIAI-AAI)*, pages 458–463, 2021.

[14] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia. Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer. *Procedia Comput. Sci.*, 179:423–431, 2021.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[16] H. Shad, M. Rizvee, N. Roza, S. Hoq, M. Khan, A. Singh, A. Zaguia, and S. Bourouis. Comparative analysis of deepfake image detection method using convolutional neural network. *Comput. Intell. Neurosci.*, 2021, 2021.

[17] S. R. Shah, S. Qadri, H. Bibi, S. M. W. Shah, M. I. Sharif, and F. Marinello. Comparing inception v3, vgg 16, vgg 19, cnn, and resnet 50: A case study on early detection of a rice disease. *Agronomy*, 13:1633, 2023.

[18] R. Shao, T. Wu, and Z. Liu. Robust sequential deepfake detection. *ArXiv*, abs/2309.14991, 2023.

[19] M. Wang and X. Gong. Metastatic cancer image binary classification based on resnet model. *2020 IEEE 20th Int. Conf. Commun. Technol. (ICCT)*, pages 1356–1359, 2020.

[20] B. Ä. Yildiz and B. GÃűkberk. A survey of deepfake detection methods. *2023 31st Signal Process. Commun. Appl. Conf. (SIU)*, pages 1–4, 2023.

[21] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang. Istvt: Interpretable spatial-temporal video transformer for deepfake detection. *IEEE Transactions on Information Forensics and Security*, 18:1335–1348, 2023.

[22] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang. Istvt: Interpretable spatial-temporal video transformer for deepfake detection. *IEEE Trans. Inf. Forensics Secur.*, 18:1335–1348, 2023.

[23] S. Zhao, H. Han, S. Shan, and X. Chen. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2185–2194, 2021.