Deepfake Analysis with EfficientNetV2 and Diffusion-Based Classification

Muhammad Amzar Syahmi **Adam Muhaimin** **Mohd Aliff Azizi**
Department of Data Science, Hanyang University
Seoul, South Korea

Abstract

The rapid advancements of generative deep learning models in recent years has led to the proliferation of realistic synthetic media, or deepfakes that proves to be difficult for the human eye to differentiate. With such a problem it has become apparent that there is an increasing need for robust deepfake detection systems capable of minimizing the threat that deepfakes poses within the context of widespread misinformation in social media. This paper presents a novel approach to deepfake detection through transfer learning of a fine-tuned EfficientNetV2 model into a diffusion based classification system that leverages the strengths of diffusion models in an attempt to create a robust detection model capable of accurately identifying deepfakes comparative to actual media, providing a sense of security for users online.

1 Introduction

Deepfakes, synthetic media generated through advanced generative deep learning models, have seen a significant rise in prevalence. These deepfakes are becoming increasingly realistic as shown in the example imagery below:

Figure 1: Sample Deepfakes by stylegan [9]

The hyper-realistic fake images produced by these deep-learning generative models poses a threat to the online community on various levels:

- Misinformation: Those with malicious intentions can spread misinformation through deepfakes, breaking trust between the public and media sources
- Propaganda: Deepfakes can be used to persuade individuals towards certain political parties by producing involvement of political leaders in fake scenarios

- Social Manipulation: Deepfakes can be used to manipulate public opinion regarding certain influential individuals or institutions thus damaging their credibility and reputation.
- Fraud: Fraudulent scams can be harder to detect as deepfakes are able toimpersonate individuals to a realistic degree

In order to combat this, we need a system that is able to detect and expose deepfake , particularly focusing on facial imagery in this project. Challenges that we faced include:

- Data Bias: Training data can be biased and the model might struggle to generalize to different scenarios (angle of face, lightning, face shapes) conditions, and face shapes.
- Time Complexity Inefficiency: Since the images are high resolution, it might take a long time for the model to train and then be used for testing
- Minor Details: Small details within the images can be a giveaway for the humannaked eye but become difficult to process for the model

This paper aims to develop a robust and efficient artificial intelligence model capable of accurately identifying deepfakes, particularly focusing on facial imagery with great generalizability towards novel deepfake media.

2 Related Work

From our literature review, many researchers have conducted deepfake classification using CNN architecture models. Here are several prominent CNN models that we found:

1. ResNet-101 [8]: ResNet-101 (Residual Neural Network) introduces an identity shortcut connection that skips one or more layers. This shortcut applies identity mapping, and the results are added to the outputs of the stacked layers. The assumption is that building a stack of identity mappings can achieve similar results as a shallow structure, allowing the network to train deeper models without degradation issues. However, research shows that this model requires high complexity and longer running times, and it can also cause gradient vanishing.

2. InceptionV3 [4]: InceptionV3 aims to improve performance regarding both computational cost and accuracy. It employs a sparsely connected network architecture, where an inception block performs convolutions on an input using multiple filter sizes (1x1, 3x3, and 5x5) within the same layer, concatenating all outputs into a single output vector. The 1x1 filter reduces dimensionality before applying another layer, hence decreasing computation costs and the number of parameters. However, research indicates that this model is resource-intensive, requiring high computational resources, and it does not generalize well across different datasets.

3. XceptionNet [5]: XceptionNet is an improved form of InceptionV3, replacing inception modules with depthwise separable convolutions followed by pointwise convolutions. This architecture applies a stack of depthwise separable convolution layers with residual connections independently over each input data channel, projecting the channel output through a depth-wise convolution into a new channel space. However, this model also requires high computational resources and training complexity. Additionally, depthwise separable convolutions add overhead, which can slow down training and inference times compared to simpler models.

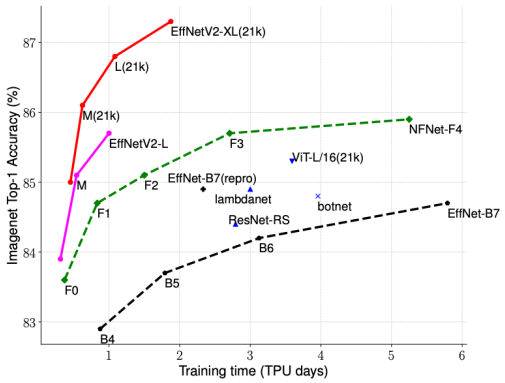
4. InceptionResNetV2 [3]: InceptionResNetV2 incorporates the benefits of both Inception and ResNet models. The inception model excels in learning features at different resolutions within the same convolution layer, while the ResNet model supports deeper CNN architectures without compromising performance. However, this architecture has high resource demands, the model size is large, and the training process is complex.

In conclusion, most of the models reviewed have shown good results, but a common limitation mentioned in the literature is the high computational cost required by these models.

3 Proposed Method

With our limited computational resources and time limitations, training a model from scratch seem to be out of the question however there are many publicly available pre-trained models that could

serve as the basis for our diffusion based classification model. For this paper we have decide to use EfficientNetV2 [10] model, known for its efficiency and scalability alongside it's high performance with less parameters than popular pre-trained models like ResNet and ViT and CARD-inspired [6] diffusion based classification layer



(a) Training efficiency.

	EfficientNet (2019)	ResNet-RS (2021)	DeiT/ViT (2021)	EfficientNetV2 (ours)
Top-1 Acc.	84.3%	84.0%	83.1%	83.9%
Parameters	43M	164M	86M	24M

(b) Parameter efficiency.

Figure 2: EfficientNetV2's performance [10]

3.1 Fine-tuning EfficientNetV2

EfficientNetV2 serves as the backbone of our diffusion based classification model. Initially, the model is pre-trained on either ImageNet21K or ImageNet1K depending on the specific version of the model. For this paper we have decided to use the EfficientNetV2-B0 version as it is best suited for our computational resources. Once loaded the model is then carefully fine-tuned with a large-scale dataset consisting of 140k images. We conducted hyper-parameter tuning to the model to ensure it best learns the characteristics of the data.

3.2 Noise Addition

After the model has been fine-tuned to our initial dataset, we added noise to our secondary dataset of 10K images. This dataset will be used to train the diffusion-based classification layer. To introduce variability and ensure robustness, we only iterated through half of the dataset. For each each image, we generate the unique parameter α_t uniformly from the interval $[0.1, 0.9]$:

For $i = 1, 2, \ldots, \lfloor \frac{N}{2} \rfloor$:

$$\alpha_t \sim \text{Uniform}(0.1, 0.9) \tag{1}$$

The corruption process is based on Gaussion noise:

$$\tilde{\mathbf{x}} = \sqrt{\alpha_t}\mathbf{x} + \sqrt{1 - \alpha_t}\mathbf{z} \tag{2}$$

where:

- $\tilde{\mathbf{x}}$ is the corrupted data.
- $\mathbf{x}$ is the original data.
- $\mathbf{z} \sim \mathcal{N}(0, 1)$ is the Gaussian noise with mean 0 and standard deviation 1.
- α_t is the noise level.

Here, $\tilde{\mathbf{x}}_i$ represents the corrupted version of the original image $\mathbf{x}_i$. This process ensures that each image is corrupted with a unique noise level controlled by α_t.

3.3 Transfer Learn into Diffusion Layer

In order to integrate our custom DiffusionLayer() into the fine-tuned model, we have frozen the layers of the model besides the classification layer which we replaced with DiffusionLayer() that has the following components:

```
class DiffusionLayer(nn.Module):
    def __init__(self, input_dim, hidden_dim, num_classes, num_steps):
        super(DiffusionLayer, self).__init__()

        encoder
            Reduces the spatial dimensions of the noisy images,
            effectively compressing the image data while preserving

        decoder
            Reconstructs the reduced spatial dimensions,
            attempting to restore the features of the original
            images from the compressed representations

        classifier
            Processes the reconstructed images and assigns labels
            to distinguish between real and fake images

    def forward(self, x):
        for step in reversed(num_steps):
            gradually decrease noise of image through encoder
            and decoder then pushing the denoised image
            through the classifier
```

This custom layer leverages the reverse diffusion process which denoises the image. This is typically how generative diffusion models learn to generate such highly realistic synthetic media. For our purposes however, we push the learned features into the classifier rather than the generator. We trained only the DiffusionLayer() with the dataset at a low epoch so that the model can familiarize itself with the mix of corrupted and original photos.

3.4 Unfreezing and Re-training

All the layers in the model are then unfrozen and retrained at a higher epoch to allow the entire model to adapt to the noise and reverse diffusion process, this effectively combines all the learned features and creates a robust detection model.

4 Results

During the initial fine-tuning process of the EfficientNetV2-B0 model, we had an outstanding validation accuracy of over 99 percent. This shows us signs of over-fitting as the model did not perform as well for the test dataset, which was generated with images from various sources excluding the training dataset. To prevent this, we implemented early stoppage into the training process. By doing so we were able to significantly reduce the risks of over-fitting.

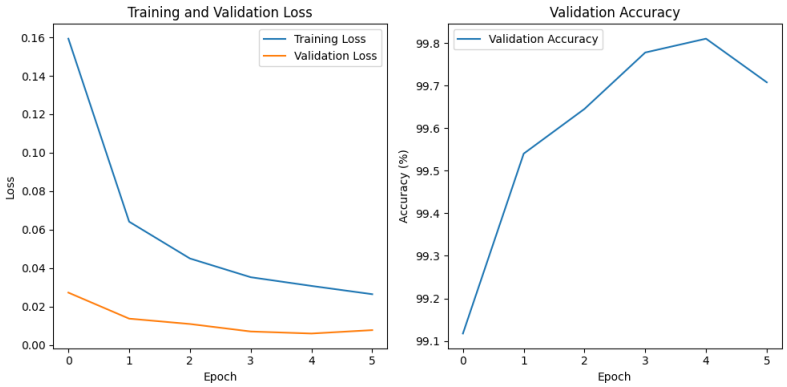


Figure 3: loss and accuracy graph of fine-tuning

Epoch	Train Loss	Validation Loss	Runtime(seconds)
1	0.1593	0.0272	1282
2	0.0641	0.0137	887
3	0.0450	0.109	915
4	0.0352	0.0070	960
5	0.307	0.0059	1020
6	0.264	0.0077	1081

Table 1: Fine-tuning results of EfficientNetV2

During the training process for the unfrozen DiffusionLayer() the model was able to adapt well to the corrupted data, achieving a relatively high validation accuracy of 89 percent. The model is generalizing quite well to noisy images.

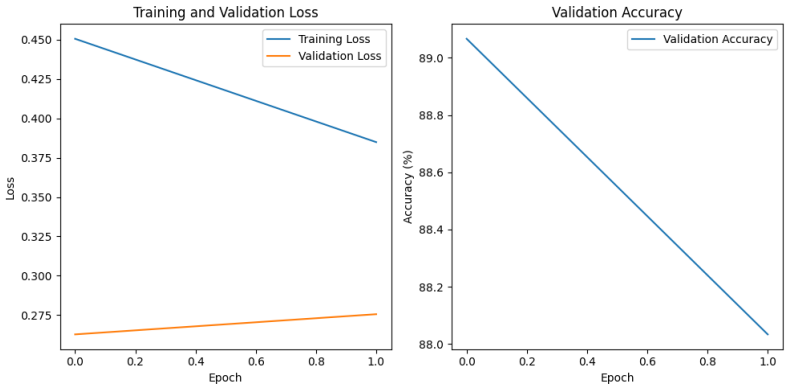


Figure 4: loss and accuracy graph of DiffusionLayer()

Epoch	Train Loss	Validation Loss	Runtime(seconds)
1	0.4506	0.2627	185
2	0.3849	0.2755	171

Table 2: Training results of DiffusionLayer()

Furthermore, we trained the model again with all of it's layers unfrozen. We actually saw a decrease in performance to 87 percent as the model's previous layers were not designed to process noisy data.

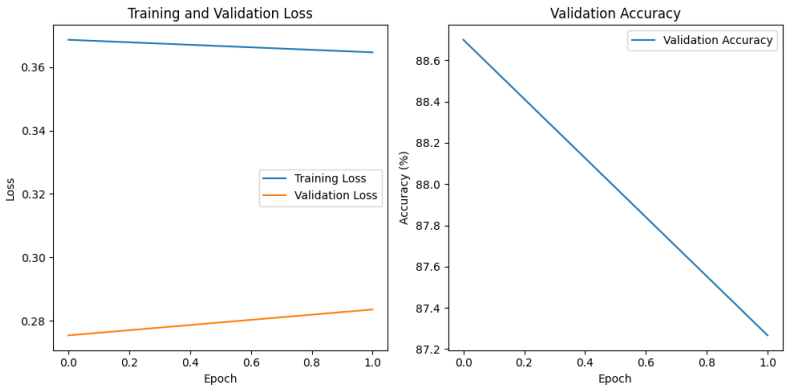


Figure 5: loss and accuracy graph of DiffusionLayer()

Epoch	Train Loss	Validation Loss	Runtime(seconds)
1	0.4506	0.2627	185
2	0.3849	0.2755	171

Table 3: Training results of unfrozen model

We tested the model with the test dataset and achieved the following:

Min-Accuracy	Max-Accuracy	Average Runtime(seconds)
0.65	0.77	541

Table 4: test dataset performance

5 Discussion

We successfully leveraged EfficientNetV2-B0 model with a CARD-inspired diffusion classifier to distinguish between real and fake images. This combination has shown a promising result, however, there are still room for improvements in the accuracy aspect, primarily due to the disparity that exist between the observed train and test accuracy.

To further address over-fitting, we proposed several strategies that were not able to be implemented due to either time, availability and memory constraints:

1. **Data Augmentation**: Technique such as AugMix and CutMix can enhance the robustness of our model by introducing a more complex and diverse data, allowing the model to learn more invariant features.
2. **Face-dataset pre-trained model**: Utilizing an EfficientNetV2-B0 model that was pre-trained on a specialized facial images dataset, such as the CelebA dataset can also enhance the performance. Domain-specific data enables the model to have a better understanding of the nuances and variations such as angles, facial expressions and accessories.
3. **Complex model**: EfficientNetV2 variations that has more parameters and advance architecture such as EfficientNetV2-M can offer a better accuracy by capturing more complex features in the images.

6 Conclusion

In this study, we have demonstrated the efficacy of employing the EfficientNetV2-B0 model combined with a CARD-inspired diffusion classifier. While our approach was able to identify real and fake images well, it is far from being considered state-of-the-art. The mentioned proposed improvements, such as applying data augmentation, using a more complex model and face-dataset pre-trained model,

can address the generalization issues our model is facing and further enhance the final accuracy. As deepfake technology is rapidly advancing with increasingly sophisticated methods to generate highly realistic images, the necessity to keep up by continually improving the model is very much needed.

In short time, our model may not be able to keep up with new deepfake images, the same fate that has already happened to the previous models we have researched on, and it is likely that our model will face the same challenge. Therefore, for future work, we want to make a model that can self-learn and adapt to maintain generalization capabilities even long after it has been created. If this can be achieved, it can serve as a safeguard against the threats posed by deepfakes in this digital age.

7 Appendix

For access to the code alongside updates relating to implantation of discussed future works discussed, refer to the GitHub repository:

EfficientNetV2-Diffusion-Classification

For access to the pre-trained EfficientNetV2-B0 model:

timm library's EfficientNetV2-B0

For access to training dataset:

140K

10K

References

References follow the acknowledgments in the camera-ready paper. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent.

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, *Inception-v4, inception-resnet and the impact of residual connections on learning*, in Proceedings of the AAAI Conference on Artificial Intelligence, 2017, vol. 31, no. 1.

[4]C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.

[5] F. Chollet, *Xception: Deep learning with depthwise separable convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251-1258.

[6] Han, X., Zheng, H., Zhou, M. (2022). *CARD: Classification and Regression Diffusion Models* . arXiv (Cornell University). https://doi.org/10.48550/arxiv.2206.07275

[7] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

[8] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[9] Karras, T., Laine, S., Aila, T. (2018). *A Style-Based Generator Architecture for Generative Adversarial Networks* . arXiv (Cornell University). https://doi.org/10.48550/arxiv.1812.04948

[10] Tan, M., Le, Q. V. (2021). *EfficientNetV2: Smaller Models and Faster Training* . arXiv (Cornell University). https://doi.org/10.48550/arxiv.2104.00298