Midway Report - Deepfake Facial Imagery Detection

Alex Zhang, Apple Jin, Alex Xiang

1. Methodology

During our research and review of previous works, we found that DenseNet121 was widely used to address similar deepfake face detection problems. According to a kaggle prize-winning solution to a deepfake detection task, its DenseNet121 attained a 0.987 accuracy in discriminating GAN generated face images. However, DenseNet121 is a large and complex model with over 8.1 million parameters. Given the need of deepfake detection technology on portable devices to address the urgent need to safeguard individuals from the proliferation of misleading digital content in real-time, we employed a transfer learning approach using the MobileNet convolutional neural network (CNN) architecture for our project. MobileNet is a lightweight (with over 4.2 million parameters) but efficient CNN model designed for mobile and embedded vision applications[1]. Its structure is shown in Table below. Despite its compact size, it has demonstrated competitive performance on various computer vision tasks, making it a suitable choice for our deepfake detection problem.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

MobileNet Model Structure

[1] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

The MobileNet model was pre-trained on the ImageNet dataset[2], which consists of over 1.2 million natural images spanning 1,000 classes. This process allowed the model to learn rich and generalizable feature representations from a diverse set of images. To adapt the pre-trained MobileNet model for deepfake detection, we performed the following steps:

01. Data Preparation:

    We used the 140k Real and Fake Faces dataset found on Kaggle by XHLULU[3]. The dataset was divided into training (50000 deepfake and 50000 real), validation (10000 deepfake and 10000 real), and test sets (10000 deepfake and 10000 real), ensuring a balanced distribution across the two classes. The detailed breakdown of this distribution is presented in the table below.

02. Transfer Learning:

    We replaced the final classification layer of the pre-trained MobileNet model with three new layers: a Global Average Pooling layer, a dense layer with 1024 neurons, and a dense layer with two output nodes, corresponding to the real and deepfake classes. The first 18 layers of the base model were frozen, preserving the initial learned feature representations from the ImageNet pre-training. We unfreezed the last 10 layers and fine tuned them to learn and generalize on our deepfake dataset.

03. Fine-tuning:

    The model was fine-tuned with a smaller learning rate on our deepfake detection dataset. During fine-tuning, the weights of the newly added classification layer and 10 unfreezed layers were updated, while the weights of the frozen layers remained fixed.

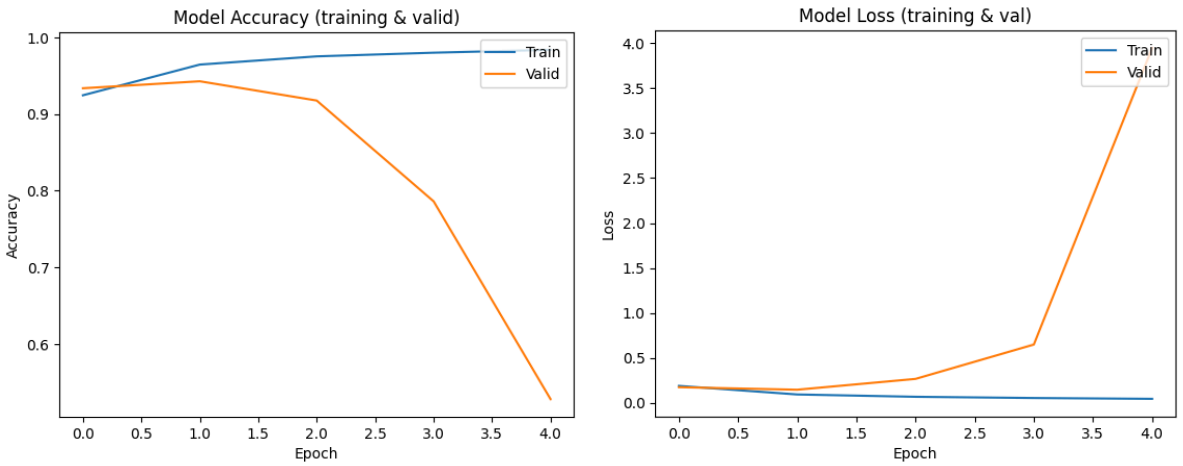
04. Training:

    After training our model with the Adam optimizer with a learning rate of 0.00001 and a batch size of 64 for 5 epochs, our fine tuned model attained a training accuracy of 0.984, while achieving a higher validation accuracy of 0.9428. We retained the weights of the epoch that has the lowest validation loss, as we can see in the below training and validation accuracy and loss curves that the model starts to overfit heavily after that.

---

[2] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[3] Xhlulu. (2020, February 10). 140k real and fake faces. Kaggle. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces/data



05. Evaluate the generalizability:

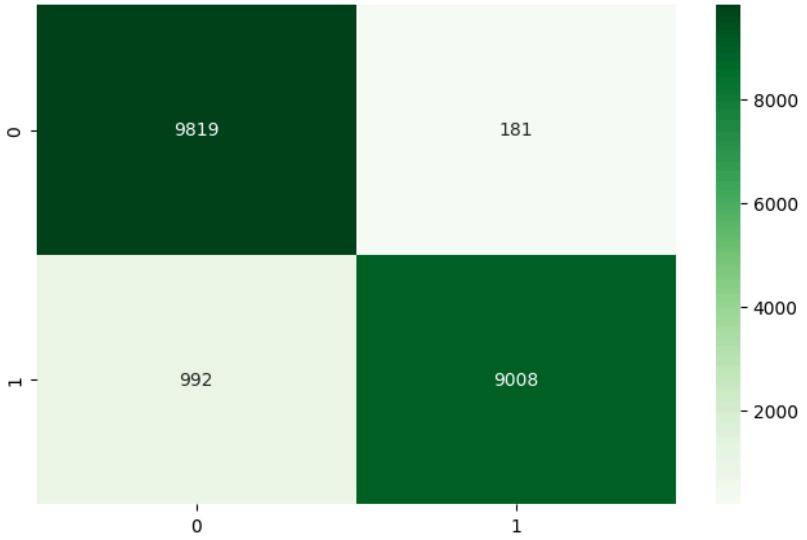
Since ideally, the model should be used to detect deepfake image generated by different model effectively, we want to know the generalizability of our model on other datasets. Thus, we obtained 3 new datasets containing deepfake images to assess the generalizability of our model.

	Cats Faces	Human Faces 1	Human Faces 2[4]
Fake	54 synthetic images created by DALL·E Mini[5]	289 synthetic images created by Stable Diffusion v1.4[6]	960 synthetic images created by GAN technology
Real	1 genuine photograph of Apple's cat	11 authentic celebrity photos randomly selected from a facial recognition dataset[7]	1081 genuine human portraits

[4] University, C. @ Y. (2019, January 14). Real and fake face detection. Kaggle. https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection
[5] OP, M. (2022, August 26). Ai Cat and dog images dall·e mini. Kaggle. https://www.kaggle.com/datasets/mattop/ai-cat-and-dog-images-dalle-mini
[6] BwandoWando. (2022, September 10). Face dataset using stable diffusion V.1.4. Kaggle. https://www.kaggle.com/datasets/bwandowando/faces-dataset-using-stable-diffusion-v14
[7] {Jha}, A. A. (2019, November 15). LFW - People (Face Recognition). Kaggle. https://www.kaggle.com/datasets/atulanandjha/lfwpeople/data

2. Result and Analysis

With this best_weight we retained, the precision of our model is 0.9803, recall is 0.9008, and F1 score is 0.9389.



Confusion matrix (fake: 0, real: 1)

According to the confusion matrix, out of the 20000 testing images with equal proportion of real and deepfake images, 9819 fake images and 9008 real images were correctly predicted by our model. 992 images that capture true human faces were predicted as AI generated images (FN), which resulted in a lower recall.

This result aligns with our expectations. Through transfer learning and fine-tuning the MobileNet model on our deepfake detection dataset, we sought to harness the pre-trained capabilities of the model while tailoring it to the nuances of our specific task. However, given that MobileNet is less complex than DenseNet 121, with fewer parameters, a modest drop in performance was anticipated. Thus we anticipated that our approach would yield satisfactory performance gains while also leveraging the computational efficiency of the model, thereby rendering it appropriate for use in resource-limited devices and settings.

To assess the generalizability of our model, we employ the saved best_weights (with lowest validation loss) from the training phase to evaluate each dataset, with the results illustrated in the table below.

	Cat Faces	Human Faces 1	Human Faces 2
Accuracy	0.0182	0.0833	0.5326
Precision	0.0182	0.0384	0.5326
Recall	1.0	1.0	0.9584
F1-score	0.0357	0.0741	0.6847

The table clearly indicates that our model's performance on each dataset is suboptimal, as evidenced by its inability to detect fake images, regardless of the distribution between real and fake image. We come up with several possible explanations include:

01. Model Overfitting:

    While the training and validation accuracy suggest that the model did not overfit significantly on the original dataset, it's possible that the model has overfit to some extent, leading to poor generalization performance on the new dataset.

    Our hypothesis posits that simplifying the model could lead to decreased accuracy on the training data but improved generalizability to new data. To test this, we trained another MobileNet with the same architecture but froze all existing layers, permitting only the newly added layers to be fine-tuned. This revised model attained an accuracy of 0.9026 on the test set during the training process. However, upon reevaluation using Human Face Dataset 2, it exhibited an accuracy of 0.5306, which is nearly identical to that of the original model.

    The fact that the simplified model did not show improved performance on the new dataset suggests that the issue might not solely be overfitting.

02. Distribution Shift:

    The significant performance drop could be attributed to a distribution shift between the original training dataset and the new evaluation dataset. The new dataset, especially the synthetic cat images and human face generated by Stable Diffusion, likely contains different characteristics, artifacts, or patterns in images distinct from the original dataset.

    Even for the human face dataset that included fake images that also generated by GAN technique, different GAN architecture can also introduce unique artifacts, flaws, or patterns in the generated images, which the model has not been exposed to during training.

03. Limited Training Data Diversity:

    Give that there can be distribution shift in different dataset, it's possible that the original training dataset lacked sufficient diversity in terms of the types of deepfake images, subjects, or generation techniques represented. If the training data was limited to a specific subset of deepfake characteristics, the model may struggle to generalize to unseen variations introduced by the new GAN architecture.

3. Plan for additional analysis:

As previously mentioned, MobileNet achieves a commendable accuracy of 94% on the primary dataset, but its performance drops significantly when evaluated on external datasets including images from different generative sources such as Stable Diffusion, GAN and StyleGAN. This difference in performance highlights the need for in-depth analysis to understand the underlying factors that influence the model's behavior on different datasets.

We speculated about limitations that AI-generated images may exhibit, such as lighting and texture incongruities, or anomalous details that deviate from human-generated norms. This curiosity extends to the mechanisms inherent in our algorithms' ability to distinguish AI-generated images from human-generated images. Specifically, the question is: what aspects of the image do we analyze to draw conclusions? Is the focus primarily on facial features, or do background elements and overall image composition play an important role?

For this reason, we propose the implementation of Grad-CAM[8] as a strategic tool for elucidating the decision-making process of the model. It will allow us to identify specific image regions and features that are prioritized by the model during the decision-making process. This is particularly important in light of the observed differences in performance, as it will allow us to determine whether the model's decisions are based on meaningful image attributes or whether they are unduly influenced by artifacts of a particular dataset.

By applying Grad-CAM, our goal is to generate a detailed visual analysis that reveals whether the model's attention is appropriately distributed across relevant features of the image or disproportionately focused on specific aspects[9], such as facial features, which may not be a primary indicator of the content generated by the AI. This analysis is expected to reveal the operational limitations of the model and guide improvements to its training scheme to increase its generalizability and effectiveness in accurately classifying images on a wider range of datasets.

Furthering our analytical journey, if Grad-CAM's insights fail to bridge the gap in generalization capabilities, our next course of action will involve augmenting our dataset with a broader spectrum of AI-generated images from varied sources like Stable Diffusion, GAN, and Midjourney. This expansion aims to ascertain whether the generalization difficulty persists across models trained on these enriched datasets. Should the expanded training sets exhibit improved generalizability, it would prompt a deeper investigation into the attributes that render certain datasets more conducive to broader applicability. Conversely, if the challenge of generalization remains ubiquitous, it will necessitate the conceptualization of a novel algorithmic approach, meticulously designed to mitigate these generalization constraints to the fullest extent possible. This iterative process of analysis and adaptation will

---

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

[9] Reiff, D. (2022, May 12). *Understand your algorithm with grad-cam*. Medium. https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353

not only refine our model's performance but also contribute significantly to the broader understanding of AI-generated content and its detection.

4. **Revised Work Plan** :

Week	Task	Details
1-2	Project Planning and Dataset Collection	**Team Meeting:** Define project goals, roles, and milestones. **(Alex Z, Alex Xiang, Apple Jin)** **Dataset Collection:** Collect a balanced dataset of AI-generated and human-taken images. Ensure diversity in the types of images gathered to cover various scenarios and artifacts. **(Alex Z, Alex Xiang, Apple Jin)**
3-4	Data Preprocessing and Model Setup	**Data Preprocessing:** Clean and preprocess the data. This includes resizing images, normalizing pixel values, and augmenting the dataset to increase its diversity. **(Alex Z, Apple Jin)** **Model Setup:** Initialize the DenseNet-121, MobileNet using TensorFlow. Set up the training and validation framework. **(Alex Xiang, Apple Jin)**
5-6	Model Training, obtaining additional data, and Initial Evaluation	**Model Training:** Begin training the models on the collected dataset. Monitor performance and adjust hyperparameters as necessary. **(Alex Z, Apple Jin)** **Obtaining additional data:** Looking for more data, and Evaluating the performance of our model on the other similar dataset. **(Alex Z, Alex Xiang, Apple Jin)** **Initial Evaluation:** Conduct an initial evaluation using accuracy, precision, and recall metrics for the classification task.**(Alex Z, Alex Xiang, Apple Jin)**
7	Model Refinement and Extended Evaluation	**Model Refinement:** Refine the models based on initial evaluation feedback. This may involve further tuning of hyperparameters or adjustments to the model architecture. **(Alex Xiang, Apple Jin)** **Extended Evaluation:** Expand the qualitative evaluation to include more images and potentially involve more domain experts and user study participants to validate the effectiveness of highlighted areas.**(Alex Z, Alex Xiang, Apple Jin)**
8	Final Evaluation, Documentation, and Presentation Prep	**Final Evaluation:** Conduct a comprehensive final evaluation of the models, focusing on both quantitative metrics and qualitative feedback. **(Alex Z, Alex Xiang)** **Documentation:** Compile detailed documentation of the project, including the methodology, model architecture, evaluation results, and insights gained.**(Alex Xiang, Apple Jin)** **Presentation Preparation:** Prepare a presentation to showcase the project's objectives, methodology, findings, and implications.**(Alex Z, Apple Jin)**