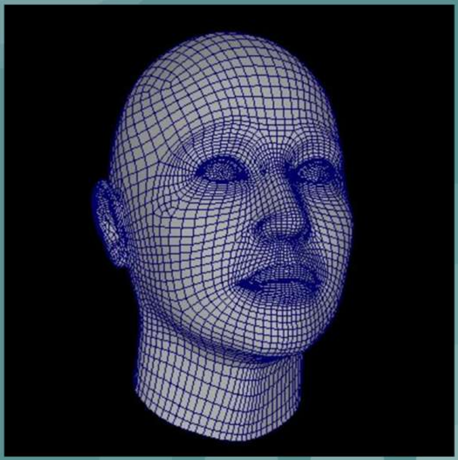
Advanced CNN Architectures for Deep fake Detection



Presented by : Shobhit Dixit

CS 615 – Deep Learning

Introduction to the Problem

- The rise of AI-generated deep fakes makes it harder to distinguish real from fake images, posing risks to security and digital authenticity.
- As deep fakes become more convincing, they contribute to misinformation, identity fraud, and privacy concerns.
- To address this, we explore different CNN architectures, comparing grayscale vs. color images, single vs. multiple kernels, and varying network depths.
- A reliable deepfake detection model can help prevent fraud, enhance security, and support forensic investigations, ensuring AI-generated content is used responsibly.

Background & Related Work

- *Faceforensics++: Learning to detect manipulated facial images.*
    - Dataset consists of 1,000 videos and trains on the single frames of each video (1.8 million images)
    - Tested binary classification for each frame, using human participants to establish a baseline
    - Comparison of 5 CNN architectures for performance on forgery detection
- *Adversarial Robustness in Deepfake Detection: Explored adversarial attacks on deepfake detectors*
    - DeepFake Detection Challenge Dataset (DFDC)
    - Proposed methods to evade detection using white-box and black-box attacks

Data Source

- We obtained our dataset from the 140k Real and Fake Faces collection from Kaggle, which includes both real and AI-generated images.
- The dataset consists of 70k real faces from Flickr, collected by Nvidia, and 70k fake faces generated by StyleGAN.
- All images were resized to 256x256 pixels and split into training, validation, and test sets for model evaluation.
- The dataset includes CSV files for metadata, ensuring easy access to image labels and facilitating structured analysis.

Data Sample

REAL IMAGE

FAKE IMAGE

- Total Samples: The dataset consists of 140,000 images, including both real and fake faces.
- Number of Classes: There are 2 classes – Real Faces and Fake Faces.
- Class Distribution: The dataset is balanced, with 70,000 real faces and 70,000 fake faces

Approach

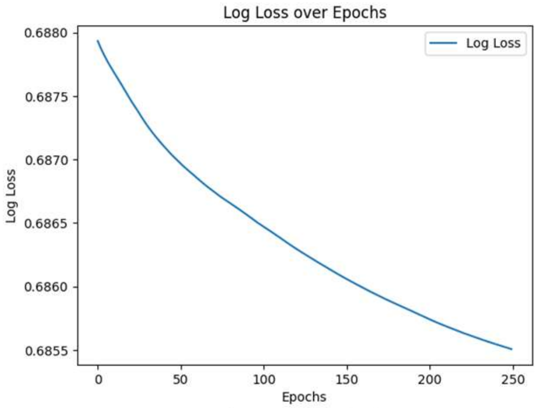
- Combined data set has a binary classification for real and fake images
- Test four separate Convolutional Neural Networks for classification accuracy
- Base Model
  - Greyscale images
  - Base architecture – Convolutional Layer > Max Pooling Layer > Flattening > Fully Connected > Logistic Sigmoid Layer
  - Evaluate with Log Loss
  - Evaluate accuracy, precision, recall, and f1
- Additional Networks
  - Same architecture as above, but using color images and a 3D kernel
  - Same architecture with Multiple kernels in the convolutional layer
  - Multiple Convolutional and Max Pooling layers prior to evaluation
- Lastly, ensemble these models to create final prediction labels for each data entry

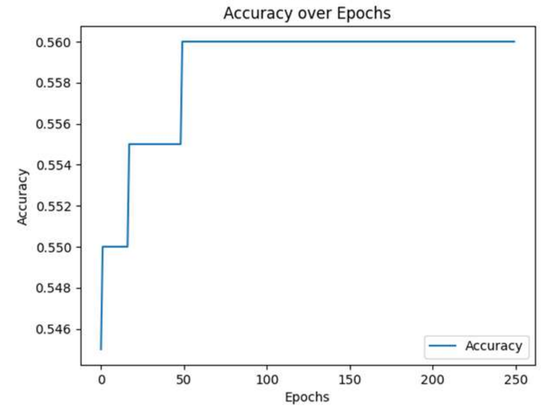
Architectures

- **Four CNN Models Tested**:
  - **Base Model**: Greyscale images, single kernel, 1 convolutional layer.
  - **Multi-Kernel CNN**: Multiple kernels for richer feature extraction. (Here we have used 3 kernels)
  - **RGB-CNN**: 3-channel inputs with 3D kernels.
  - **Multi-Layer CNN**: Stacked convolutional and pooling layers. (2 convolution, 2 pooling and 2 fully connected layers)
- **Common Framework**:
  - Convolution → Max Pooling → Flattening → Fully Connected → Sigmoid/Log Loss.

Base Model - Greyscale, 1 CNN, 1 Kernel

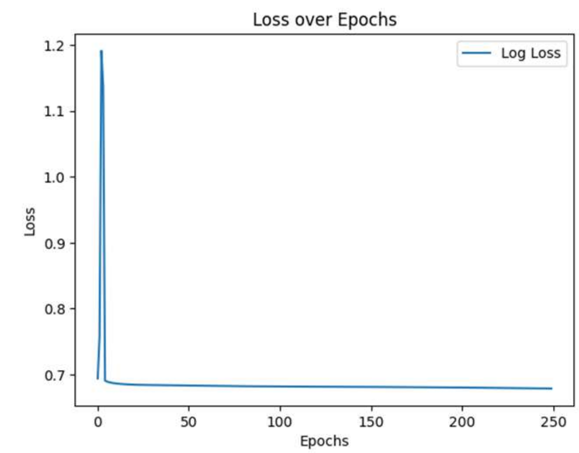
Epoch	TRUE	FALSE	Last Acc	Last 100 Epochs (in sec)
10	8.0000	192.0000	0.5500	941.868559
20	7.0000	193.0000	0.5550	
30	7.0000	193.0000	0.5550	
40	5.0000	195.0000	0.5550	
50	4.0000	196.0000	0.5600	
60	4.0000	196.0000	0.5600	
70	4.0000	196.0000	0.5600	
80	4.0000	196.0000	0.5600	
90	4.0000	196.0000	0.5600	

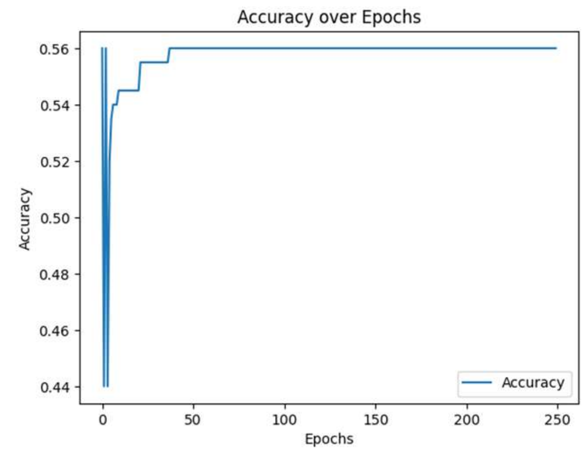




CNN WITH MULTIPLE KERNELS

Epoch	Accuracy	Loss	Last 100 Epochs (in sec)
10	0.5450	0.6865	3408.8100
20	0.5450	0.6847	
30	0.5550	0.6841	
40	0.5600	0.6838	
50	0.5600	0.6835	
60	0.5600	0.6827	
70	0.5600	0.6831	
80	0.5600	0.6823	
90	0.5600	0.6821	





CNN WITH RGB CHANNELS

<table>
<tr><th>Epoch</th><th>Accuracy</th><th>Loss</th><th>Last 100 Epochs (in sec)</th></tr>
<tr><td>10</td><td>0.5000</td><td>0.8713</td><td rowspan="9">2847.3900</td></tr>
<tr><td>20</td><td>0.5000</td><td>3.4316</td></tr>
<tr><td>30</td><td>0.5000</td><td>0.9378</td></tr>
<tr><td>40</td><td>0.5000</td><td>1.2222</td></tr>
<tr><td>50</td><td>0.5000</td><td>0.7047</td></tr>
<tr><td>60</td><td>0.5000</td><td>0.7081</td></tr>
<tr><td>70</td><td>0.5000</td><td>0.7039</td></tr>
<tr><td>80</td><td>0.5000</td><td>0.7005</td></tr>
<tr><td>90</td><td>0.5000</td><td>0.7389</td></tr>
</table>

CNN WITH MULTIPLE CONVOLUTION LAYERS

Epoch	Time for last 10 epochs (in sec)	Train Accuracy	Train Loss
10	124.9000	0.5000	0.7420
20	123.1600	0.5000	0.7386
30	123.9400	0.5000	0.7361
40	123.0300	0.5000	0.7339
50	123.0400	0.5000	0.7319
60	122.2800	0.5000	0.7300
70	123.3700	0.5000	0.7282
80	123.0600	0.5000	0.7265
90	122.6200	0.5000	0.7248

Analysis

- **Limited Training Data:** The dataset contained only 200 images, which is insufficient for training deep CNN models effectively, leading to poor generalization.
- **Shallow Architectures:** The models, particularly the Base Model and Multi-Kernel CNN, had minimal layers, restricting their ability to extract complex patterns required for deep-fake detection.
- **Feature Extraction Constraints:** The Base Model (greyscale, single kernel) lacked diversity in feature extraction, while even the Multi-Kernel CNN (with 3 kernels) struggled due to limited depth.
- **Impact of Color Information:** The RGB-CNN, despite processing 3-channel images, did not significantly outperform other models, indicating that color alone is not a decisive factor in deep-fake detection.
- **Inadequate Feature Hierarchy:** The Multi-Layer CNN, though deeper, still underperformed due to insufficient layers and limited dataset size, restricting meaningful feature extraction.
- **Overfitting & Poor Generalization:** The small dataset likely led to overfitting, causing the models to perform well on training data but poorly on unseen images.
- **High Computational Cost & Long Training Time:** Despite using only a few epochs, training was time-consuming due to convolutional operations, especially in Multi-Kernel CNN, RGB-CNN, and Multi-Layer CNN, which required higher computational resources.

FUTURE WORK

- **Increase Dataset Size** – Train on thousands of images for better generalization.
- **Use Deeper & Optimized Architectures** – We can mimic transfer learning architectures such as RESNET50, VGG16. By observing and studying these architectures in detail, we can create a similar architecture with similar depths.
- **Implement Better Regularization** – Use dropout, batch normalization, and L2 regularization to prevent overfitting.

Citations

1. NVIDIA Flickr Faces Dataset: Karras, T., Laine, S., & Aila, T. (2019). *A Style-Based Generator Architecture for Generative Adversarial Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Retrieved from https://arxiv.org/abs/1812.04948
2. 1 Million Fake Faces - StyleGAN: Karras, T., Aittala, M., Aila, T., & Laine, S. (2020). *Training Generative Adversarial Networks with Limited Data*. Retrieved from https://arxiv.org/abs/2006.06676
3. Deepfake Detection Challenge Discussion (Kaggle): Kaggle Community. (2020). *Deepfake Detection Challenge: Discussion on Real and Fake Face Datasets*. Retrieved from https://www.kaggle.com/c/deepfake-detection-challenge/discussion/122786
4. CNN Architectures for Image Classification: Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems (NeurIPS). Retrieved from https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
5. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. arXiv.org. Retrieved from https://arxiv.org/abs/1901.08971
6. Gupta, G., Raja, K., Gupta, M., Jan, T., Whiteside, S. T., & Prasad, M. (2024). A Comprehensive Review of DeepFake Detection Using Advanced Machine Learning and Fusion Methods. Electronics, 13(1), 95. Retrieved from https://doi.org/10.3390/electronics13010095

Thank you!