AI Image Classifier: Learning to Identify AI-Generated Images & Faces using Binary Image Classification

Taise Miyazumi, Tyler Wong, Joshua Kim,
Daniel Franco, Antonio Villarreal
CIS4930: Paper Presentation

Introduction

AI Image Classifier

- **Image Generation Technology**
  - Advanced models like Dall-E and ChatGPT are producing more realistic outputs
  - Distinguishing AI-generated content from real content is increasingly challenging
- **Potential Risks**
  - Erosion of trust in digital media
  - Potential identity theft
  - Rise in misinformation
- **Solution**
  - **AI Image Classifier -** a binary image classifier
  - Develop a convolutional neural network (CNN) to classify images as real or AI-generated
  - Utilize a labeled dataset from Kaggle for training the model

A fake image generated with StyleGan [1].

A real image [1].

Related Work

Binary Image Classification

- **What is Binary Image Classification**
  - Binary Image Classification categorizes images into one of two distinct groups
  - For our application, differentiating between real and machine-generated images
- **Deep Learning Models**
  - Deep learning models are preferred due to their ability to learn detailed features and express complex structures
- **Convolutional Neural Networks (CNNs)**
  - CNNs are highly effective for binary image classification and our choice of model
  - CNNs are trained using large datasets and sophisticated deep learning techniques to improve classification accuracy.

Methodology

Dataset

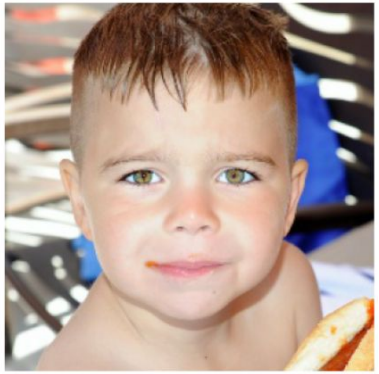
All images are resized to 224x224 during training

Image Classification Branch

- 174,427 total images
- 91,993 real images and 82,434 AI generated images
- Images range from natural and city landscapes, animals, and portraits

Real Images

AI Generated Images

Real


StyleGAN

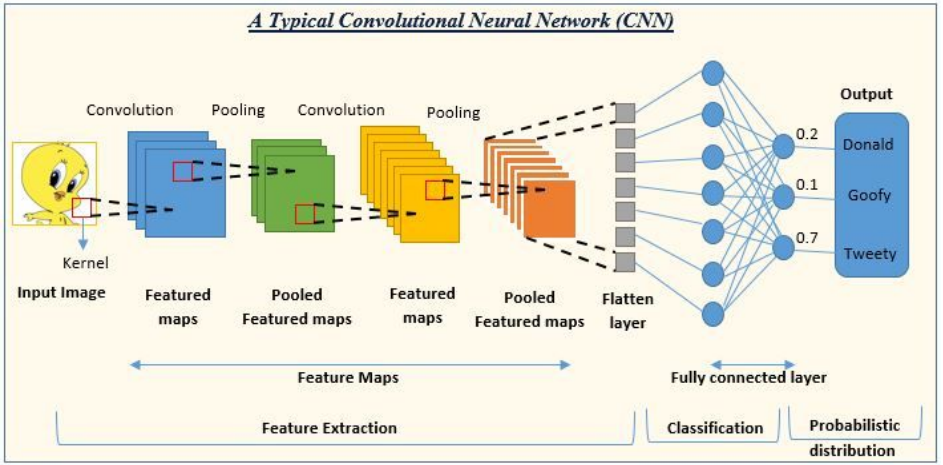
Dataset Cont.

Face Classification Branch

- 140,000 images split 50/50 between real and fake (StyleGAN)
- Split roughly 71% for training, 14% testing and 14% validation subsets
- Total of 100,000 training images, 20,000 testing and 20,000 validation

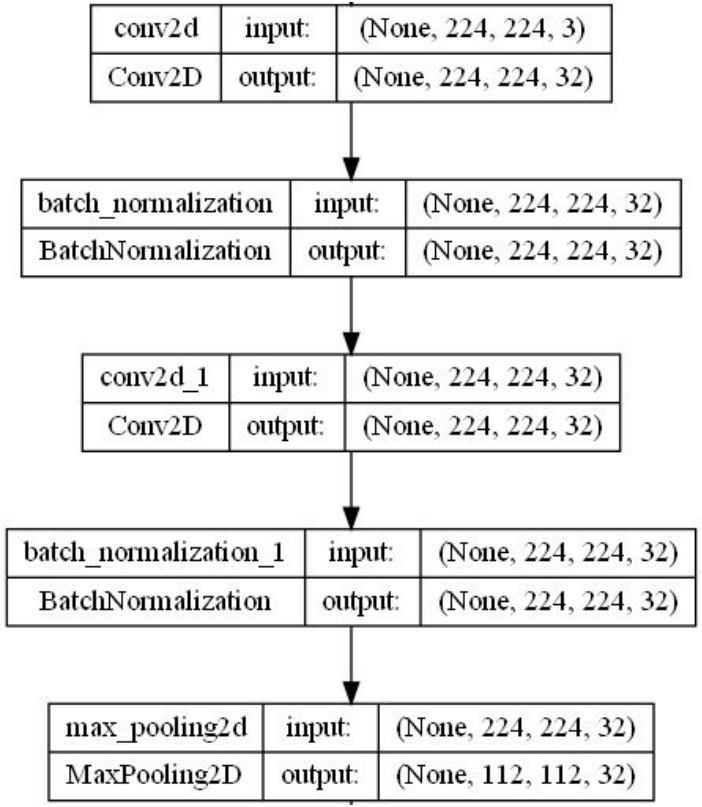
Architecture

- Convolutional Neural Network
- Selected CNN architecture due to their ability to capture spatial hierarchies
- Highly customizable
- Consists of five single convolutional blocks and a final block layer that produces output.



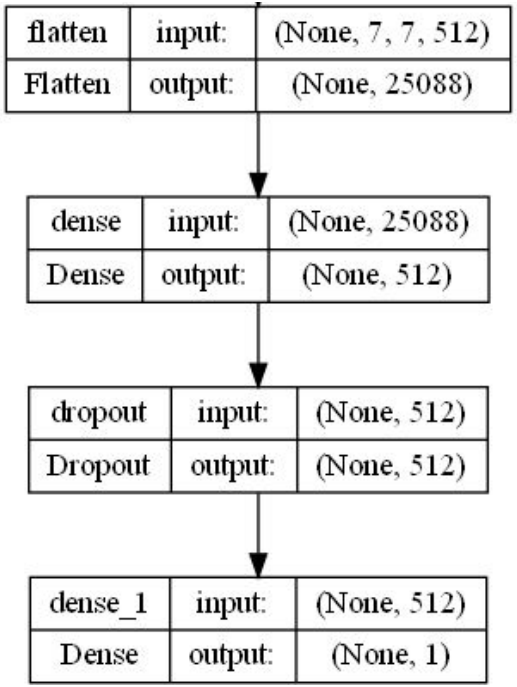
Single Convolutional Blocks

- Convolutional Layer
  - Designed to detect features within the data
  - Feature can include edges, textures, or even more complex data
- Batch Normalization
  - Standardizes the inputs for the activation functions (ReLU & Sigmoid)
  - Helps in stabilizing and accelerating the training process
- 2D Max Pooling
  - Downsample feature maps
  - Makes the model more efficient and robust to variations in the data



Final Convolutional Block

- Flatten Layer
  - Converts multidimensional output of the preceding layers to a 1D array
- Dense Layer
  - High level reasoning component of the network
  - ReLU introduces non-linearity to the model
  - Sigmoid enables binary classification
- Dropout Regularization
  - Applied right before output layer
  - Prevents overfitting and improves generalization



Experimental Results & Discussion

Hyperparams

The face classifier model was trained with the following hyperparameters:

- Batch Size: 32
- Image Size: 224x224 pixels
- Epochs: 10
- Optimizer: Adam

The image classifier model followed a similar pattern. Here are the hyperparameters for the image classifier model:

- Batch Size: 16
- Image Size: 224x224 pixels
- Epochs: 10
- Optimizer: Adam

Face Classifier Model

Loss	Accuracy	Validation Loss	Validation Accuracy
0.895644844	0.609979987	0.566042364	0.701900005
0.552488446	0.722469985	0.499188811	0.780799985
0.412680358	0.819429994	0.411865324	0.812300026
0.25865823	0.903699994	0.356503189	0.866450012
0.155344114	0.946399987	0.396919668	0.834800005
0.094690487	0.967140019	0.1005973	0.957099974
0.069315463	0.977039993	0.076931074	0.973349988
0.051306203	0.98259002	0.092519313	0.973950028
0.041600782	0.985289991	0.073294029	0.980250001
0.03494842	0.988179982	0.061438367	0.980799973

Image Classifier Model

Loss	Accuracy	Validation Loss	Validation Accuracy
0.751404226	0.657130599	0.414012462	0.803411186
0.328613222	0.856442571	0.234398171	0.899670362
0.225830555	0.900601983	0.195697546	0.913802505
0.180971742	0.919263303	0.181041405	0.919678926
0.153293461	0.929554224	0.157337084	0.928278625
0.136970967	0.935459375	0.156433761	0.930199206
0.121399291	0.941973627	0.164964303	0.930170536
0.108042113	0.947549105	0.159408733	0.926788032
0.098821238	0.953303695	0.158378646	0.934441745
0.089601584	0.956908405	0.189360917	0.93561703

Results

The results of the face classifier model:

- Final Training Accuracy: 98.82%
- Final Validation Accuracy: 98.08%
- Training Loss: 0.0349
- Validation Loss: 0.0614

The results of the image classifier model:

- Final Training Accuracy: 95.69%
- Final Validation Accuracy: 93.56%
- Training Loss: 0.0896
- Validation Loss: 0.1894

Conclusion

Conclusion

Challenges:

- Generative AI and its endless possibility
  - Human face generation
  - Image generation

Our Solution:

- 2 CNN models:
  - Face classification
  - Image classification
- Created website using streamlit and flask
  - Input our own images and returns if it is AI generated

Conclusion

What was found?

- Able to create an impressive model that detects AI generated images.

Future work

- Discovering, generating more datasets
- Ensemble techniques for higher accuracy
- Access to higher computing machines
- Deploying website and model

References

[1]“140k Real and Fake Faces,” *www.kaggle.com*. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces/data (accessed Apr. 14, 2024).

GitHub Link

https://github.com/Antonio-Villarreal/ai_classification_project