AI-generated Image Detection

Team:

Alina Al Beaini

Amanda Pan

Cemile Kurkoglu

Hasan Saad

THE ERDŐS INSTITUTE

DATA SCIENCE BOOTCAMP

Motivation

AI-generated images have become increasingly realistic, prompting a variety of malicious uses.

- e.g. misinformation, impersonating celebrities, and fake social media accounts

fake

real

Question: How to Detect AI-Created Images

Convolutional Neural Network (CNN)

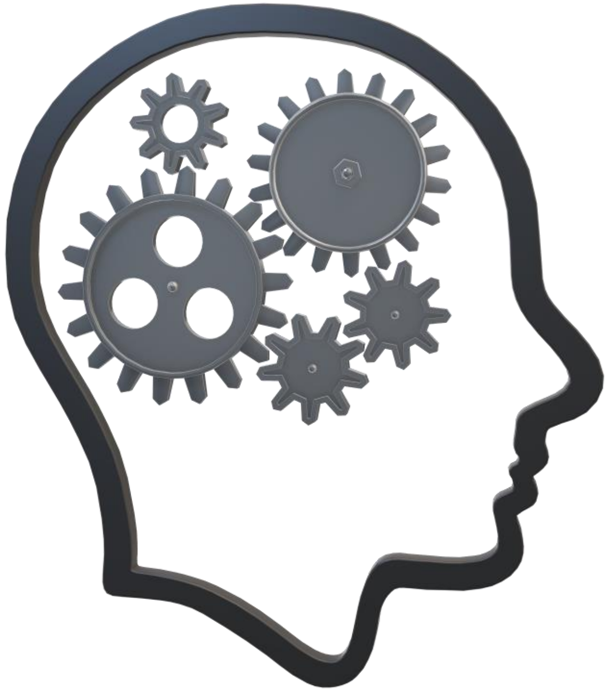
Data Gathering

Cleaning & Preprocessing

Modeling

Model Evaluation

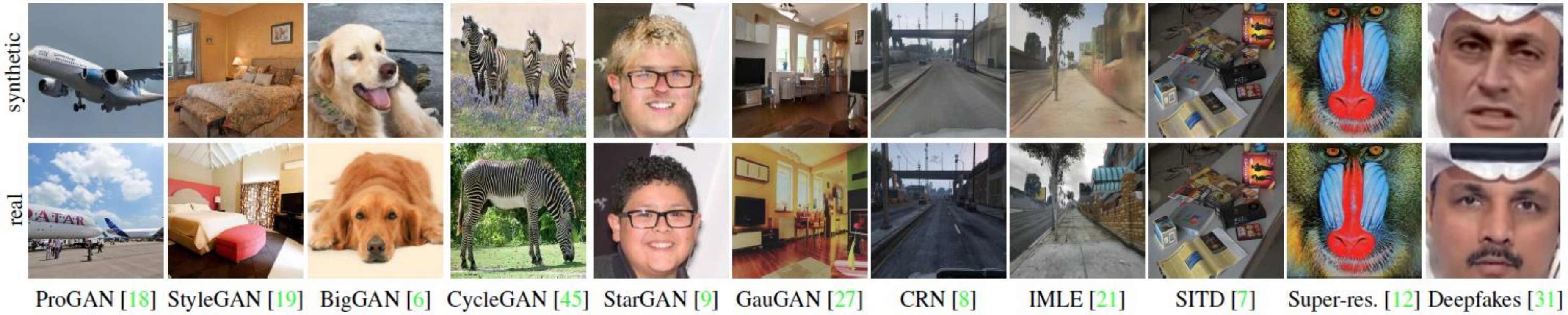
Future Directions



Data Gathering

- GitHub Repository: CNN Detection

- Our dataset consists of real and fake images generated by 13 different CNN-based image generator models (90,460 items).



Source: https://github.com/peterwang512/CNNDetection

Cleaning and Preprocessing

Studying Dataset

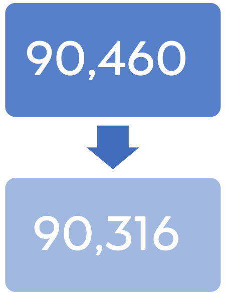
Initial Challenges: Varied dimensions, and grayscale images.

Observation: Majority of images were 256x256 pixels.

Cleaning Dataset

(1) Disregarded all images that are grayscale and of dimension less than 256x256 pixels.

(2) Cropped all images larger than 256x256 pixels.



Re-labeling Files

Included necessary output information.

"real_n" and "fake_n_generator".

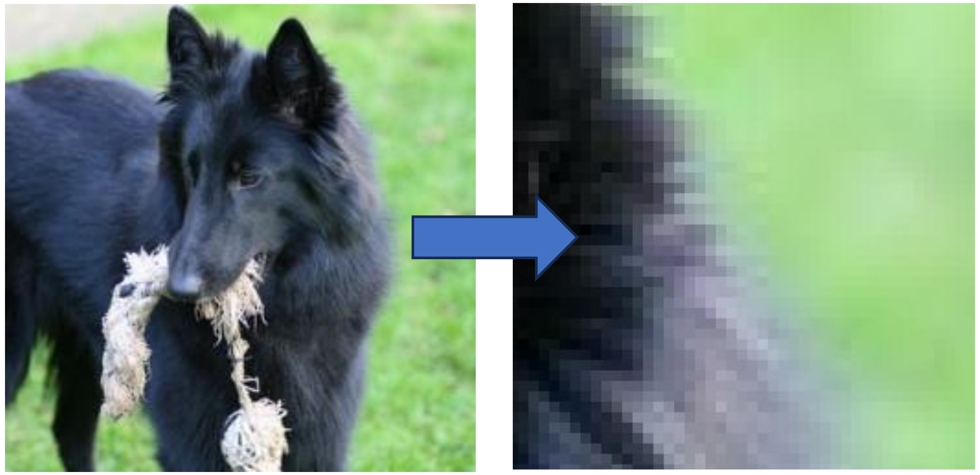
Loading Images

Initial Challenge: Loading all images into memory is impractical.

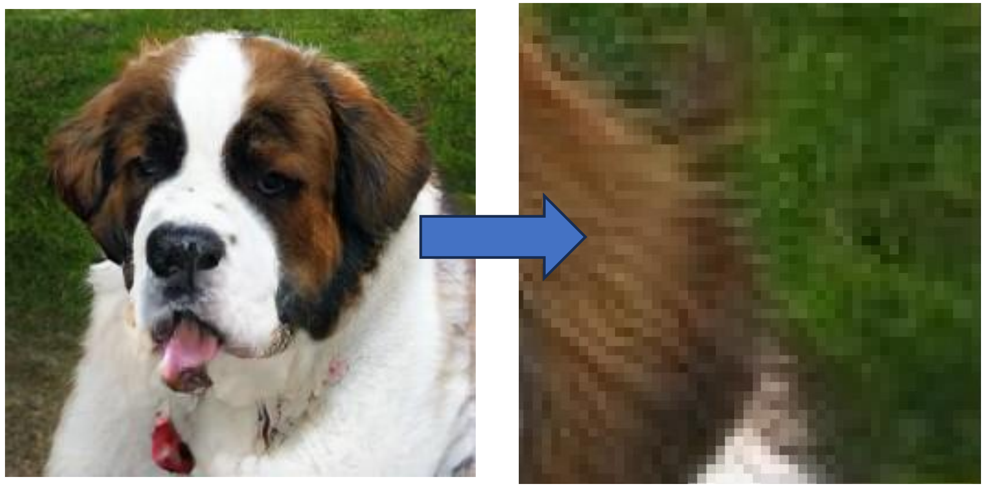
Solution: Loaded images per batches instead.

AI Fingerprints

Real photo

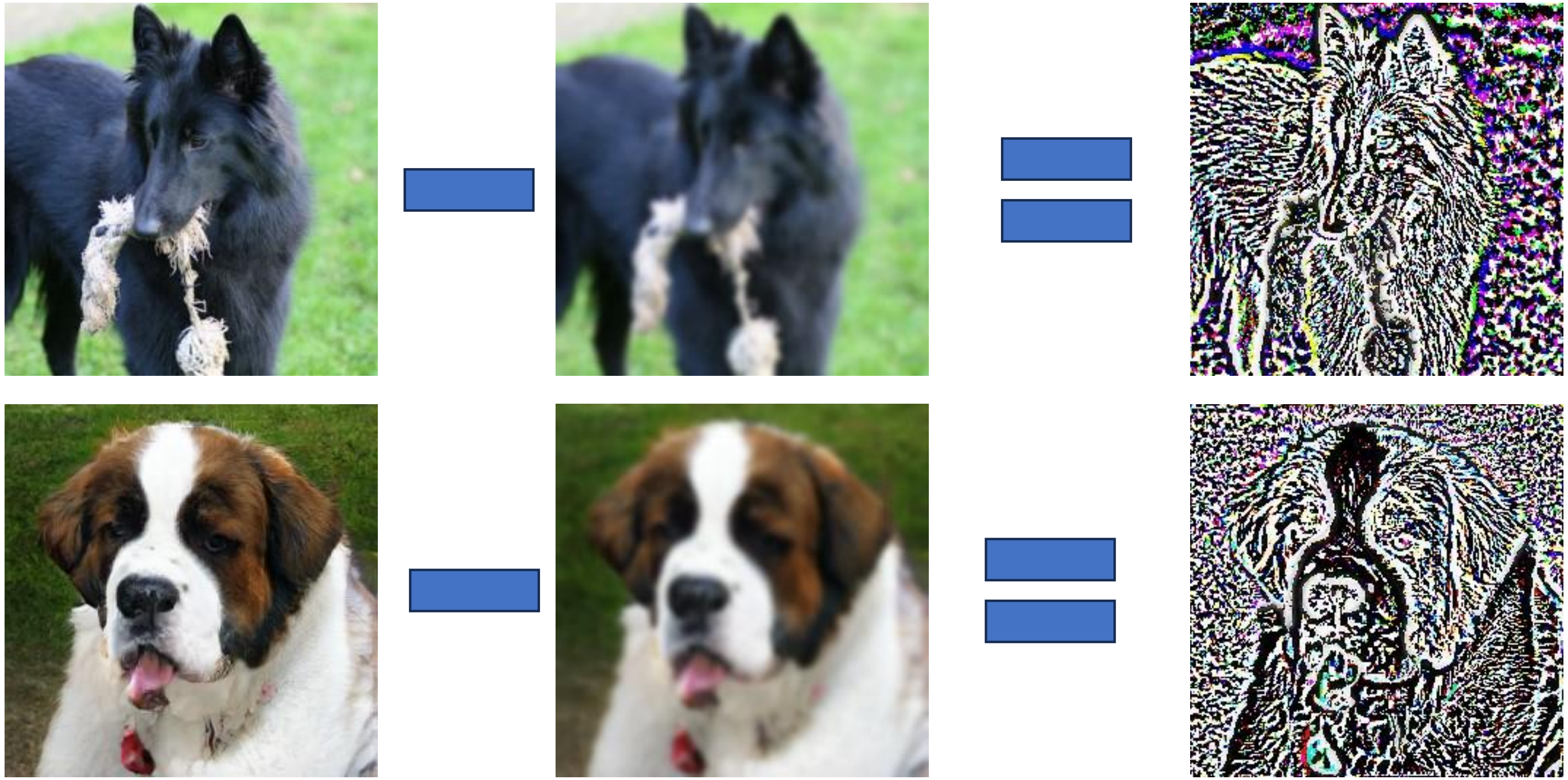


Generated
By ProGAN

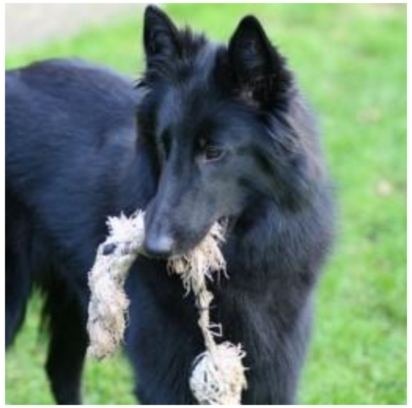


AI-generated images tend have a distinct "texture."

Gaussian High-Pass Filter

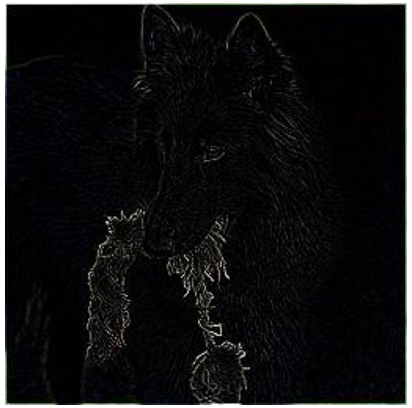


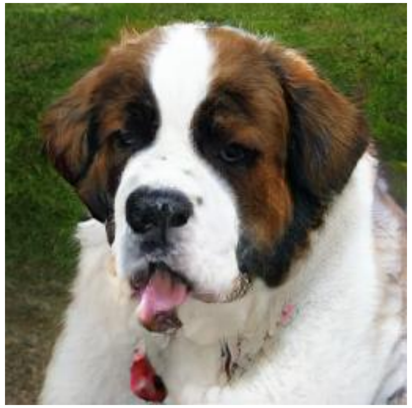
Convolutional High-Pass Filter



Each color channel is convolved with:

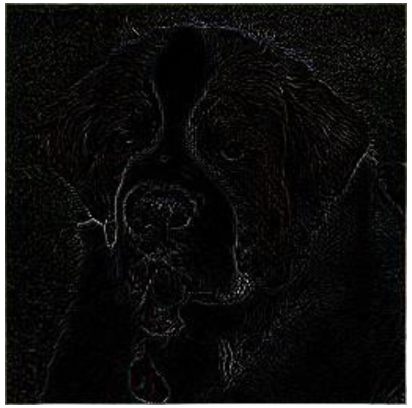
$$\star \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} =$$



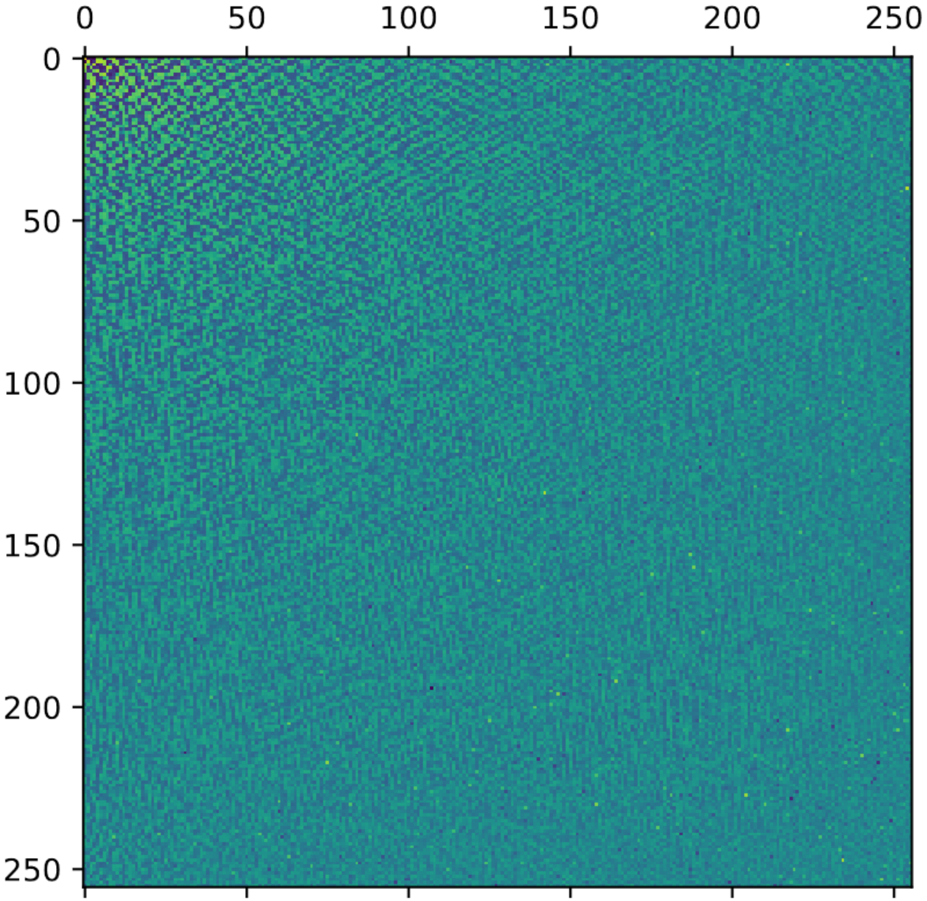


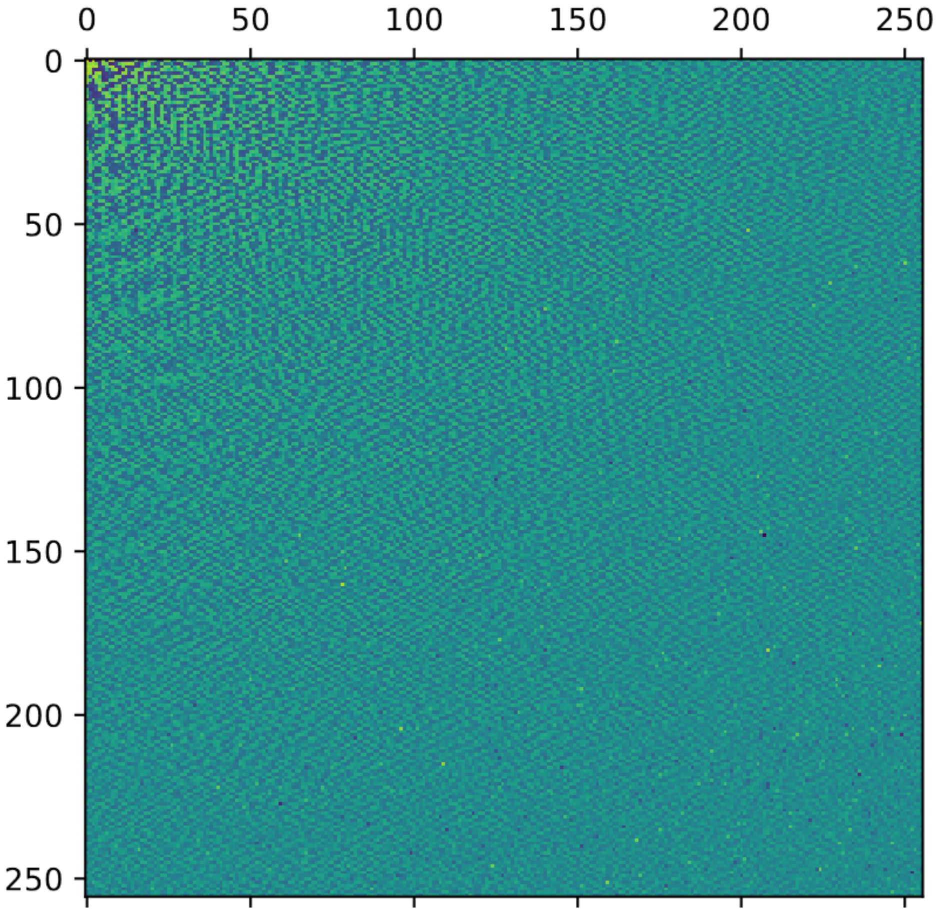
Each color channel is convolved with:

$$\star \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} =$$

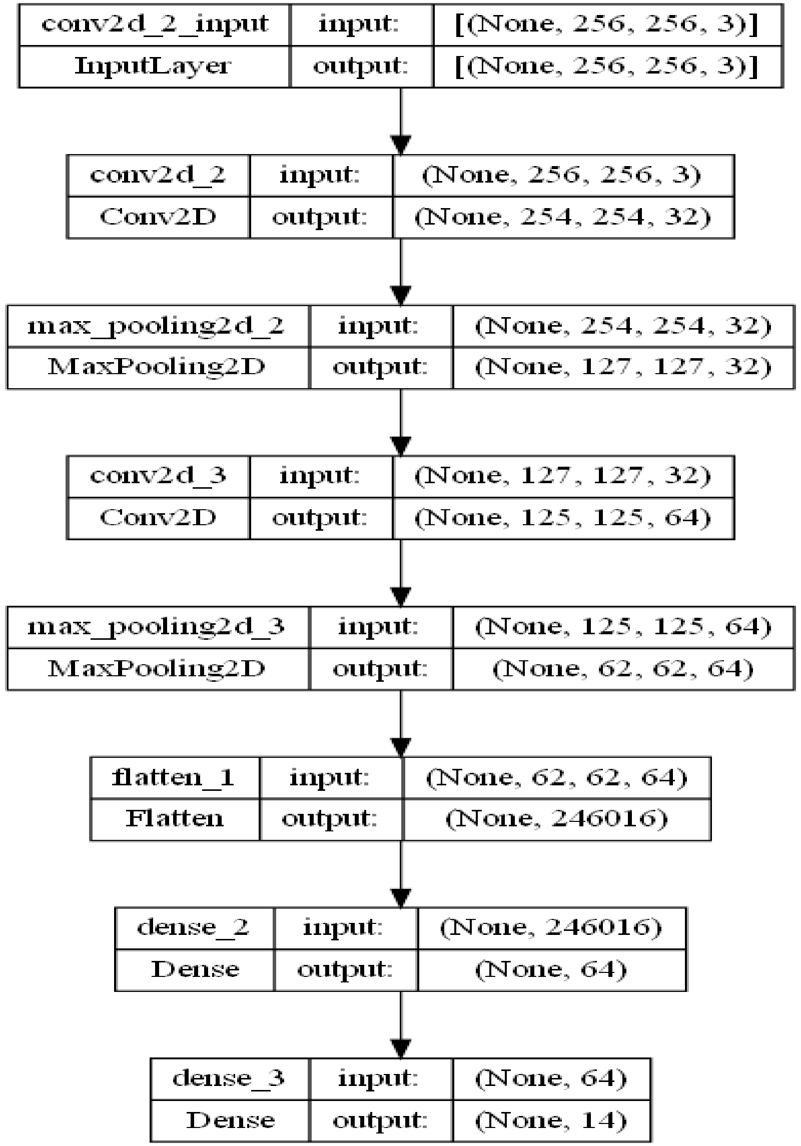


Discrete Cosine Transform





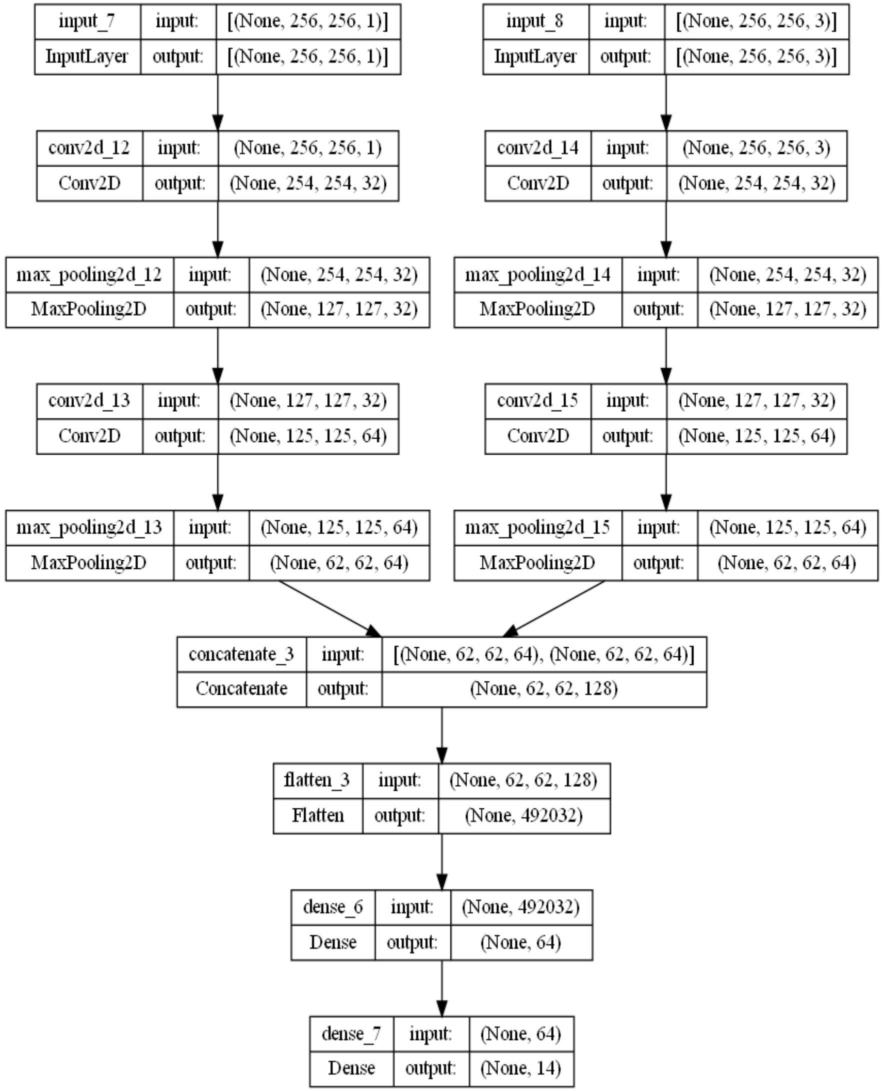
Model Architecture: Single-Channel



conv2d_2_input	input:	[(None, 256, 256, 3)]
InputLayer	output:	[(None, 256, 256, 3)]
conv2d_2	input:	(None, 256, 256, 3)
Conv2D	output:	(None, 254, 254, 32)
max_pooling2d_2	input:	(None, 254, 254, 32)
MaxPooling2D	output:	(None, 127, 127, 32)
conv2d_3	input:	(None, 127, 127, 32)
Conv2D	output:	(None, 125, 125, 64)
max_pooling2d_3	input:	(None, 125, 125, 64)
MaxPooling2D	output:	(None, 62, 62, 64)
flatten_1	input:	(None, 62, 62, 64)
Flatten	output:	(None, 246016)
dense_2	input:	(None, 246016)
Dense	output:	(None, 64)
dense_3	input:	(None, 64)
Dense	output:	(None, 14)

- The high-pass filtered image goes through input.
- Alternating convolutional and pooling layers with ReLU activation function
- One dense hidden layer with ReLU activation
- Output layer consists of 14 outputs with softmax activation that correspond to each possible AI generator.

Model Architecture: Dual-Channel



input_7 | input: | [(None, 256, 256, 1)]
InputLayer | output: | [(None, 256, 256, 1)]

input_8 | input: | [(None, 256, 256, 3)]
InputLayer | output: | [(None, 256, 256, 3)]

conv2d_12 | input: | (None, 256, 256, 1)
Conv2D | output: | (None, 254, 254, 32)

conv2d_14 | input: | (None, 256, 256, 3)
Conv2D | output: | (None, 254, 254, 32)

max_pooling2d_12 | input: | (None, 254, 254, 32)
MaxPooling2D | output: | (None, 127, 127, 32)

max_pooling2d_14 | input: | (None, 254, 254, 32)
MaxPooling2D | output: | (None, 127, 127, 32)

conv2d_13 | input: | (None, 127, 127, 32)
Conv2D | output: | (None, 125, 125, 64)

conv2d_15 | input: | (None, 127, 127, 32)
Conv2D | output: | (None, 125, 125, 64)

max_pooling2d_13 | input: | (None, 125, 125, 64)
MaxPooling2D | output: | (None, 62, 62, 64)

max_pooling2d_15 | input: | (None, 125, 125, 64)
MaxPooling2D | output: | (None, 62, 62, 64)

concatenate_3 | input: | [(None, 62, 62, 64), (None, 62, 62, 64)]
Concatenate | output: | (None, 62, 62, 128)

flatten_3 | input: | (None, 62, 62, 128)
Flatten | output: | (None, 492032)

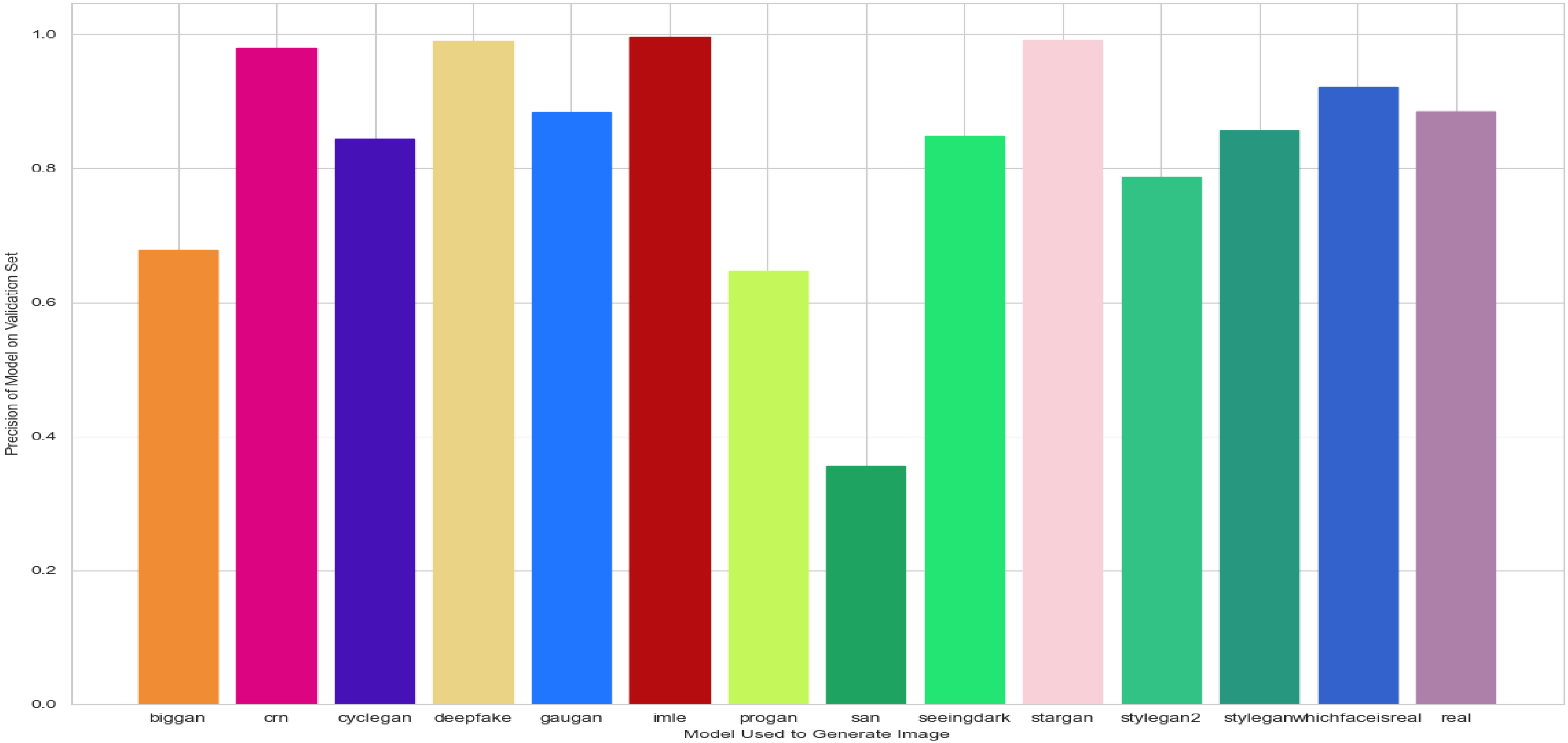
dense_6 | input: | (None, 492032)
Dense | output: | (None, 64)

dense_7 | input: | (None, 64)
Dense | output: | (None, 14)

- We start with two unconnected channels so we can benefit from the power of two filters at once.
- The first input takes the high-pass filtered image, while the second input takes the discrete cosine transform.
- Alternating convolutional and pooling layers for each channel with ReLU activation function
- Channels are merged and a dense hidden layer is added.
- Output layer consists of 14 outputs with softmax activation that correspond to each possible AI generator.

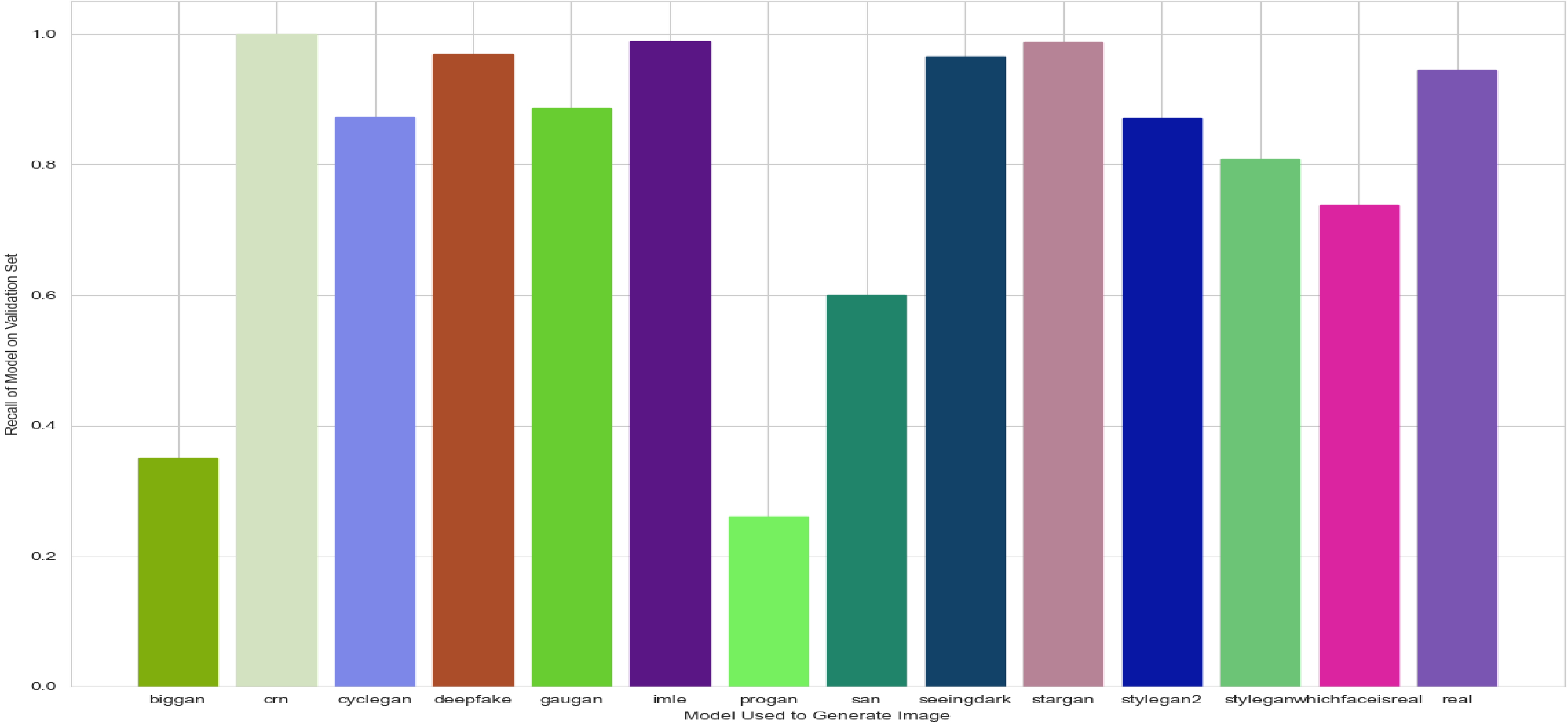
Model Evaluation: Multiclass Precision

- Multiclass Precision on Validation Sets (Dual-Channel Model):

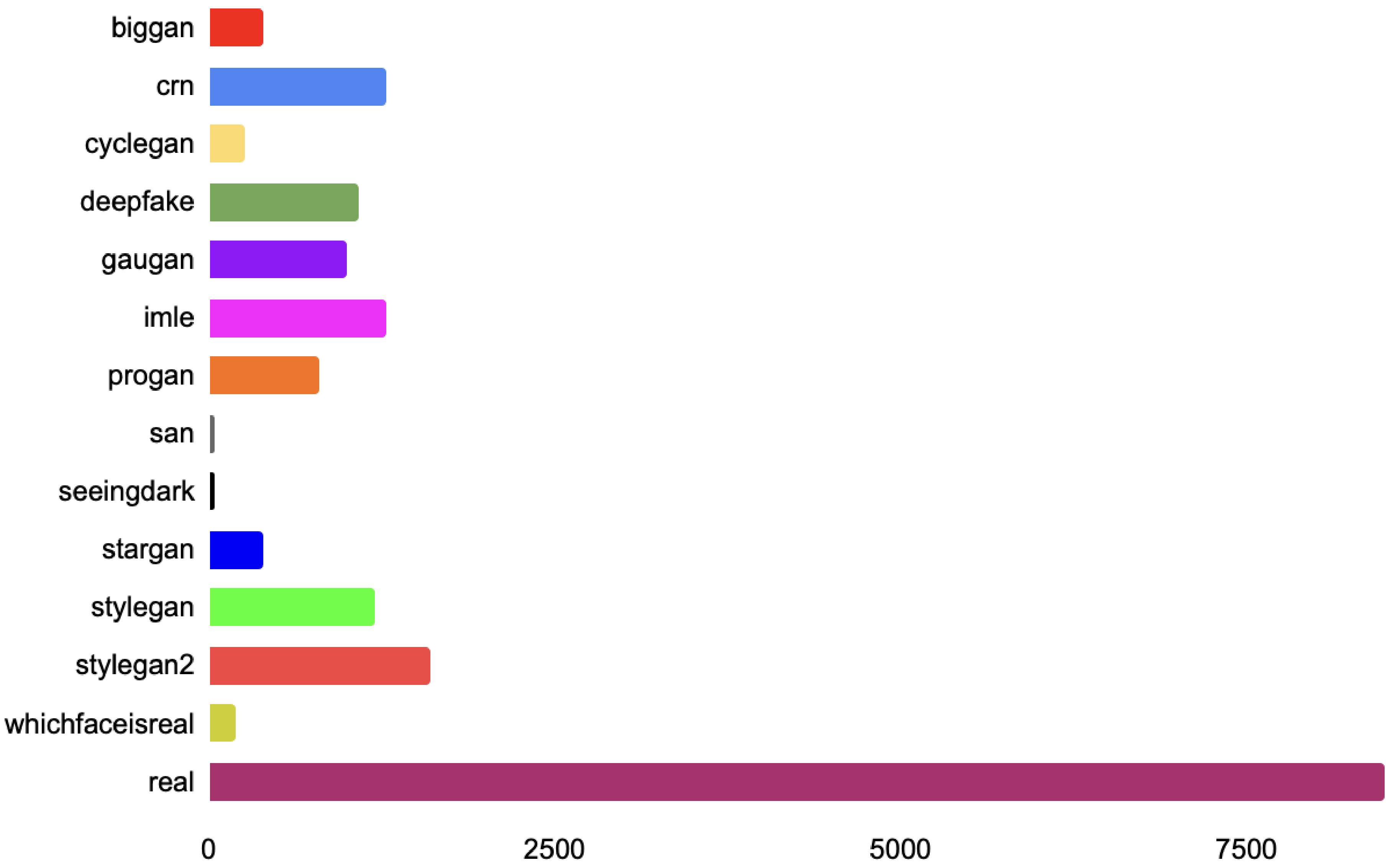


Model Evaluation: Multiclass Recall

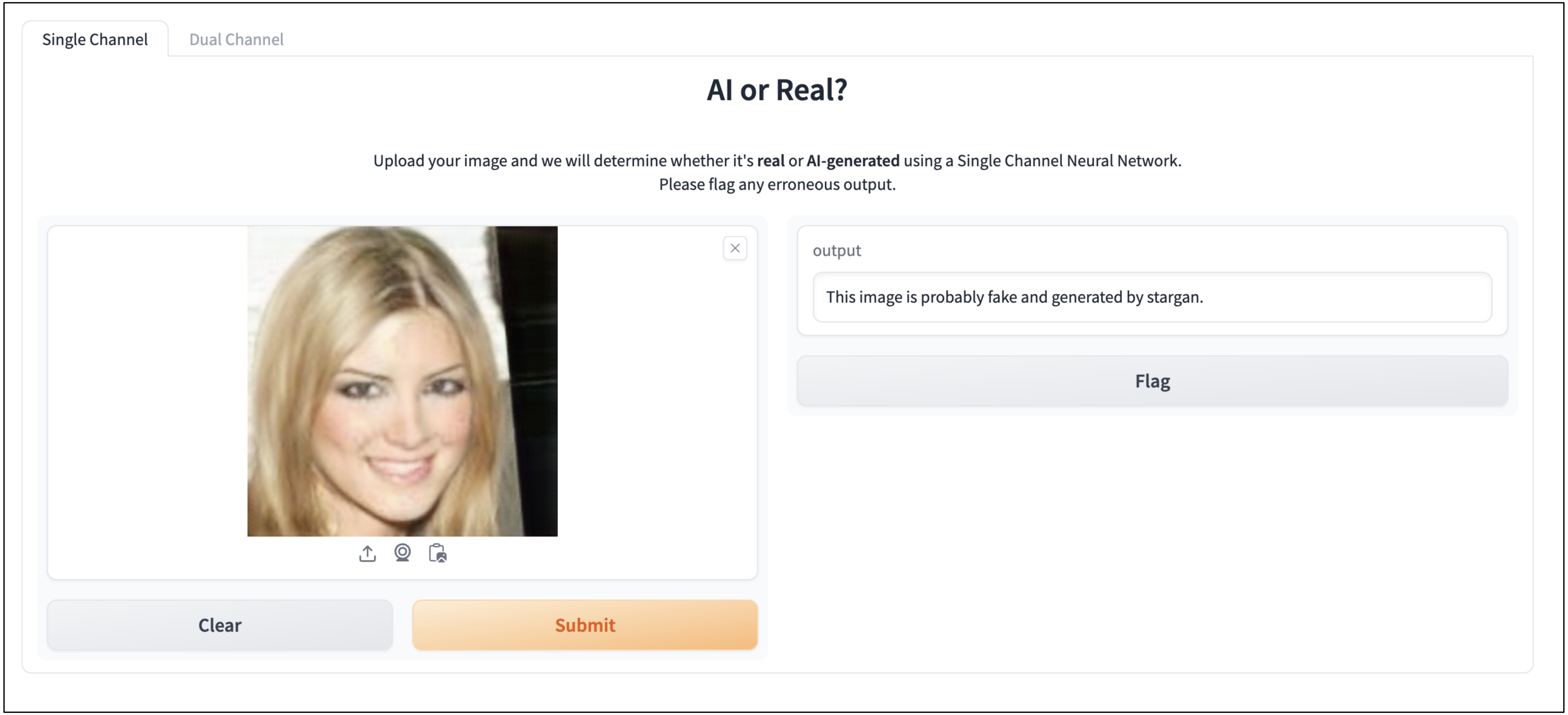
- Multiclass Recall on Validation Sets (Dual-Channel Model):



Model Evaluation



Gradio Interface



AI or Real?

Upload your image and we will determine whether it's **real** or **AI-generated** using a Single Channel Neural Network.
Please flag any erroneous output.

output: This image is probably fake and generated by stargan.

Future Directions

- Train on more images to avoid overfitting.
- Include images that are resized, compressed, or altered adversarially.
- Generalize to different methods of image generation.
- Work on a more homogeneous dataset in terms of model/method that generated the images to improve precision and recall.
- Try other filters such as co-occurrence matrices, or in different color modes.

Thank You!