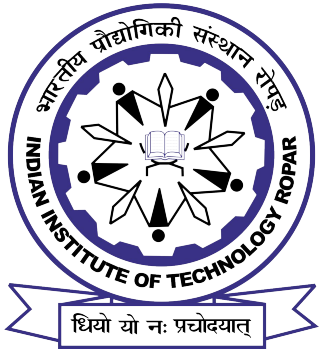
Explainable AI for Deepfake Image Analysis



Supervisor: Dr. Puneet Goyal

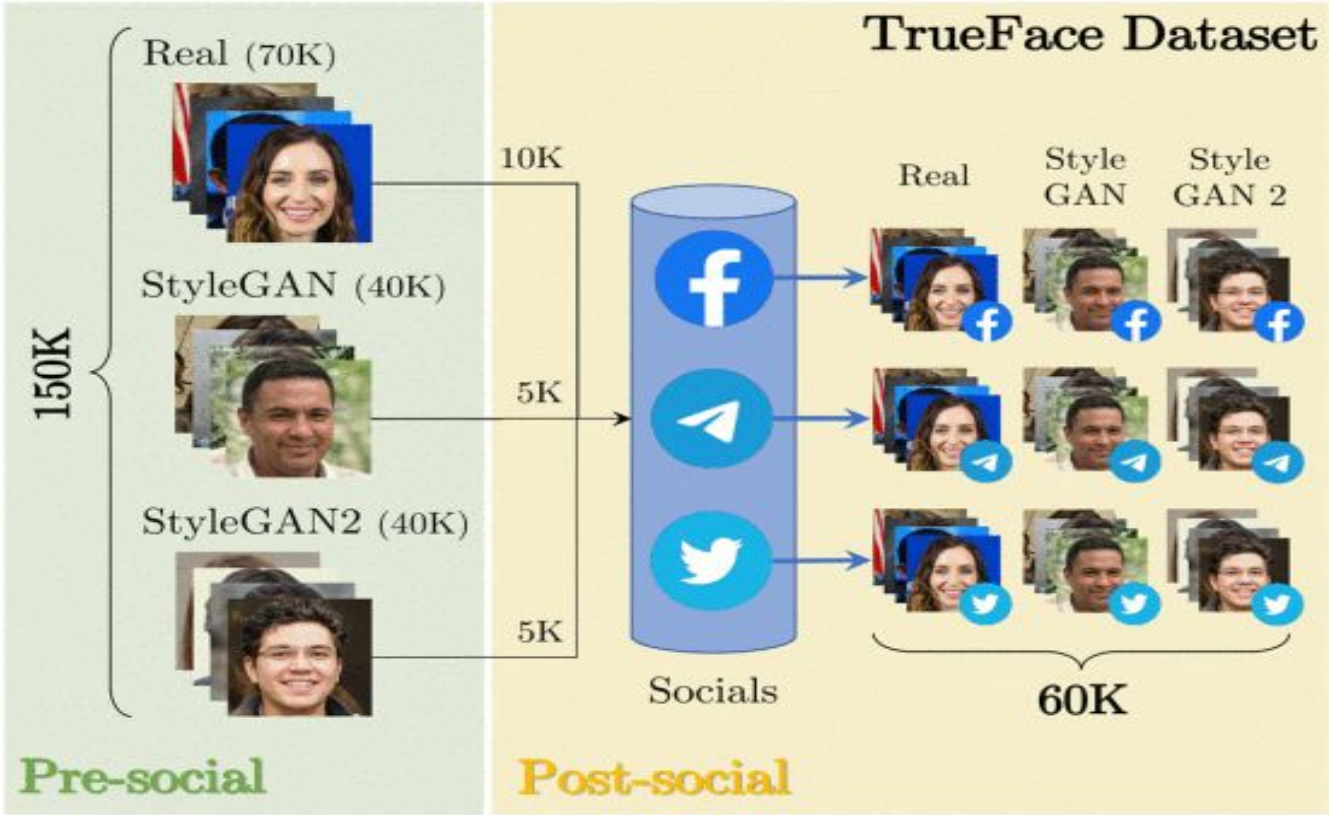
Jadhav Abhilasha S. (2022CSM1001)
Protyay Dey (2022AIM1009)

Overview

- DeepFake
- TrueFace
- An Effective CNN-based Approach for Synthetic Face Image Detection in Pre-Social and Post-Social Media Context (Accepted at 8th International Conference on Computer Vision and Image Processing (CVIP) IIT Jammu, 2023)
    - Why EfficientNet?
    - Results and Experimental Analysis
    - Ablation Study
- Why Interpretability Maps?
- Various Class Activation Maps / Interpretability Maps
- Future Directions

DeepFake?

Source: Facebook AI Launches Its Deepfake Detection Challenge At the NeurIPS conference, Facebook asked researchers to build tools to spot deepfake videos



Reference: G. Boato et al., "TrueFace: a Dataset for the Detection of Synthetic Face Images from Social Networks," *2022 IEEE International Joint Conference on Biometrics (IJCB)*, Abu Dhabi, United Arab Emirates

TrueFace Dataset

- **First publicly available** dataset for synthetic-vs-real image classification, with data shared via social networks.
- **210K face images**, divided into pre-social and post-social collections.
- The pre-social dataset includes **150K face images**, with 70K real images (from **FFHQ Dataset**) and 80K synthetic images generated by **StyleGAN** and **StyleGAN2** models (having different variations based on the fantasy parameter **ψ**).
- **ψ** determines how far the generative network should deviate from the data average. By adjusting this parameter, the variety/quality tradeoff of the output data can be determined.
- The post-social dataset consists of **60K images**, half real and half fake, uploaded and downloaded on Facebook, Telegram, and Twitter.

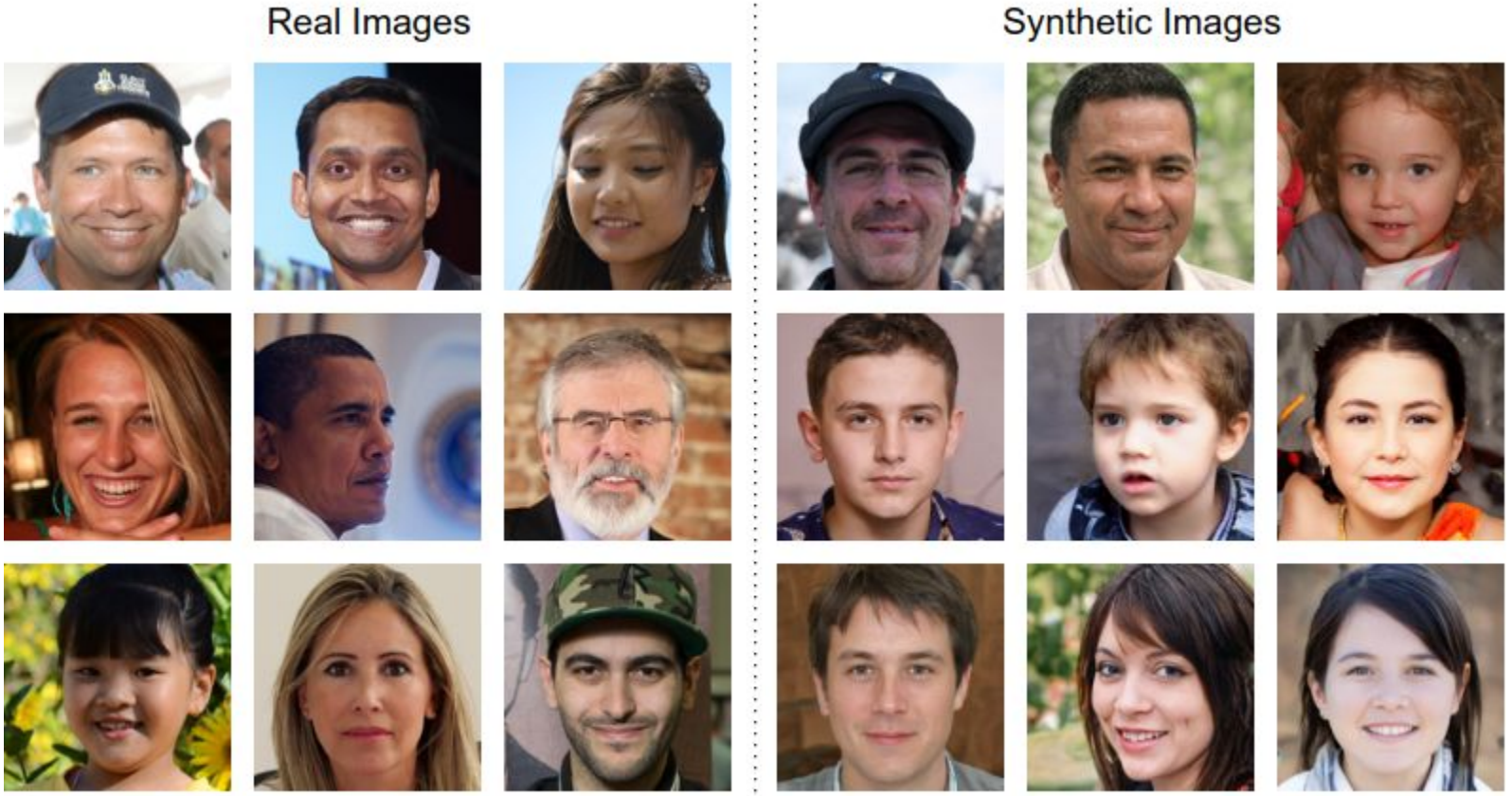
Real	**StyleGAN**		**StyleGAN2**	
	$\psi = 0.7$	$\psi = 1$	$\psi = 0.5$	$\psi = 1$
70K	20K	20K	20K	20K

Table 1. Pre-social dataset.

Platform	**Real**	**StyleGAN**	**StyleGAN2**
Facebook (FB)	10K	5K	5K
Telegram (TL)	10K	5K	5K
Twitter (TW)	10K	5K	5K

Table 2. Post-social dataset.

Reference: G. Boato et al., "TrueFace: a Dataset for the Detection of Synthetic Face Images from Social Networks," *2022 IEEE International Joint Conference on Biometrics (IJCB),* Abu Dhabi, United Arab Emirates



Reference: G. Boato et al., "TrueFace: a Dataset for the Detection of Synthetic Face Images from Social Networks," *2022 IEEE International Joint Conference on Biometrics (IJCB)*, Abu Dhabi, United Arab Emirates.

Why EfficientNet? (out of so many deep neural network architectures)

- **Fewer parameters** compared to most state-of-the-art architectures.
- Ramachandran et al. gave Swish activation function that is used in EfficientNet and in their paper they write "extensive experiments show that **Swish** consistently matches or outperforms ReLU on deep networks applied to a variety of challenging domains such as image classification and machine translation".
- Adaptability to Different Resolution
- The incorporation of **compound scaling** approach that ensures the model's depth, width, and resolution are adjusted together, EfficientNet effectively mitigates overfitting tendencies.

Results and Experimental Analysis

Models	# Parameters	Accuracy	F1-score
VGG19	144M	99.95	99.95
ResNet50	26M	99.71	99.71
DenseNet121	7.97M	99.82	99.83
MnasNet	4M	99.88	99.88
EfficientNet-B2	9.2M	**99.97**	**99.97**

Comparative analysis of different CNN based methods on pre-social set of TrueFace dataset.

Results and Experimental Analysis

Models	Pre-Social	Telegram	Twitter	Facebook	Combined
VGG19	99.09	99.90	99.92	99.87	99.73
ResNet50	97.90	99.82	99.82	99.78	99.39
DenseNet121	98.94	99.90	99.92	99.93	99.73
MnasNet	99.40	99.93	99.93	99.92	99.79
EfficientNet-B2	**99.85**	**100**	**100**	**99.98**	**99.96**

Accuracy of various models when trained over the combined (pre-social and post-social) dataset and cross-tested across different splits of TrueFace dataset.

Training dataset	Models	Pre-Social	Telegram	Twitter	Facebook	Overall
Telegram	VGG19	96.84	96.78	93.80	97.10	96.15
	ResNet50	96.68	96.43	98.65	98.75	97.65
	DenseNet121	98.40	98.18	99.17	99.25	98.75
	MnasNet	95.80	95.58	97.73	98.67	96.85
	EfficientNet-B2	**99.05**	**99.15**	**99.55**	**99.67**	**99.38**
Twitter	VGG19	93.11	97.03	97.22	97.55	96.28
	ResNet50	93.59	97.93	97.23	98.27	96.74
	DenseNet121	96.18	98.83	98.20	99.18	98.11
	MnasNet	95.49	97.93	94.82	96.38	96.09
	EfficientNet-B2	**97.97**	**99.57**	**99.17**	**99.52**	**99.10**
Facebook	VGG19	93.75	98.22	97.63	93.67	95.97
	ResNet50	95.46	98.72	98.72	95.22	97.03
	DenseNet121	98.00	99.38	99.32	97.93	98.73
	MnasNet	97.20	99.20	98.87	97.25	98.14
	EfficientNet-B2	**98.65**	**99.68**	**99.67**	**98.87**	**99.25**

Accuracy of various models trained on different datasets and tested on different social media platforms.

Ablation study

- **Superior** performance than B0 and B1 variant of EfficientNet.
- EfficientNet B2's exceptional performance highlights its robustness in capturing intricate patterns and features in images shared on platforms like Twitter, Facebook, and Telegram.

Models	Pre-Social	Telegram	Twitter	Facebook	Combined	Average
EfficientNet-B0	99.09	98.98	98.85	99.63	99.14	99.14
EfficientNet-B1	99.41	99.38	97.50	99.55	98.93	98.95
EfficientNet-B2	99.05	99.15	99.55	99.67	99.38	**99.36**

Our Contributions

- We presented a CNN-based EfficientNet-B2 model specifically designed for classifying real and synthetic facial images shared on social networks. The model included compound scaling, which allowed us to balance accuracy and computational efficiency.
- Extensive experiments were conducted on the TrueFace dataset containing real and synthetic facial images shared on popular social media platforms such as Facebook, Twitter, and Telegram.
- The robustness of our approach was evaluated in a cross-dataset environment, demonstrating that it could generalize and accurately classify real and synthetic images shared across different social media platforms.
- We performed a comprehensive evaluation of the proposed method by comparing it with alternative CNN-based approaches such as VGG19, ResNet50, DenseNet121, and MnasNet.

Few Queries that we came across

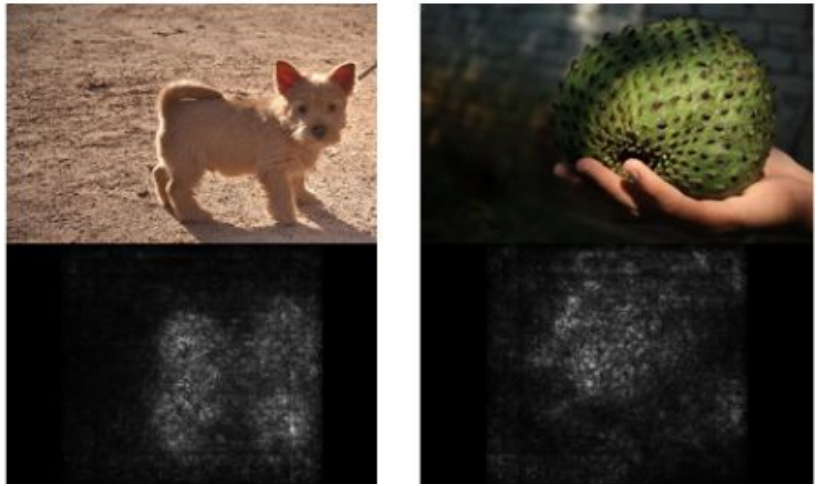
- What enables the model to achieve such high classification accuracy for images?
- What is the mechanism through which the model classifies images?
- How does the model's perception of images contribute to its exceptional classification accuracy?

Different types of Data Visualization Techniques:

- Saliency Maps
- Activation Maps
- Class Activation Maps (CAMs)
- Heatmaps
- Activation Maximization
- Grad-CAM (Gradient-weighted Class Activation Mapping)
- Guided Grad-CAM

Saliency Maps

- **Purpose:** Saliency maps highlight crucial image regions, aiding interpretation of neural network decisions and revealing which features drive predictions.
- **Method:** They're generated by calculating gradients or backpropagating through a model, creating heatmaps that prioritize salient pixels.
- **Interpretability:** Saliency maps provide insight into model reasoning, useful for debugging, model improvement, and understanding critical image regions.
- **Applications:** They're employed in object detection, image captioning, medical imaging, and more, enhancing model transparency and performance.



Source: Simonyan, K. et al. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps.

Generating Saliency Maps.

To compute the class saliency map for a given image and class these steps are followed:

1. Find the derivative w using back-propagation.
2. Rearrange the elements of the derivative vector w to create a saliency map M.
3. For grayscale images, the map is computed as $M_{ij} = |w_{h(i,j)}|$.
4. For multi-channel images (e.g., RGB), the color channel corresponding to each pixel's element in w is considered and maximum magnitude across all channels to compute $M_{ij} = \max_c |w_{h(i,j,c)}|$ is taken.

Given an image I_o, a class c, and a classification ConvNet with the class score function $S_c(I)$, we would like to rank the pixels of I_o based on their influence on the score $S_c(I_o)$.

$$S_c(I) = w_c^T I + b_c,$$

Interpretation of computing the image-specific class saliency using the class score derivative is that the magnitude of the derivative indicates which pixels need to be changed the least to affect the class score the most.

$$w = \frac{\partial S_c}{\partial I}\bigg|_{I_0}.$$

Maximum magnitude of w across all colour channels:

$$M_{ij} = \max_c |w_{h(i,j,c)}|.$$

Class Activation Maps

- **Objective:** CAMs aim to localize the most discriminative regions of an input image that contribute to a specific class prediction made by a convolutional neural network (CNN).
- **Method:** CAMs are generated by taking the weighted sum of feature maps from the final convolutional layer of a CNN, where the weights are derived from the final fully connected layer.
- **Heatmap Visualization**
- **Applications:** CAMs are valuable for tasks like object detection, where they provide not only the class prediction but also the localization of the object within an image.

Zhou, B. et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Class Activation Maps

For a given image, let $f_k(x, y)$ represent the activation of unit k in the last convolutional layer at spatial location (x, y). Then, for unit k, the result of performing global average pooling, F^k is $\sum_{x,y} f_k(x, y)$. Thus, for a given class c, the input to the softmax, S_c, is $\sum_k w_k^c F_k$ where w_k^c is the weight corresponding to class c for unit k. Essentially, w_k^c indicates the *importance* of F_k for class c. Finally the output of the softmax for class c, P_c is given by $\frac{\exp(S_c)}{\sum_c \exp(S_c)}$.

We compute a weighted sum of the feature maps of the last convolutional layer to obtain our class activation maps. Here unit is individual feature map or channel

By plugging $F_k = \sum_{x,y} f_k(x, y)$ into the class score, S_c, we obtain

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y)$$
$$= \sum_{x,y} \sum_k w_k^c f_k(x, y). \quad (1)$$

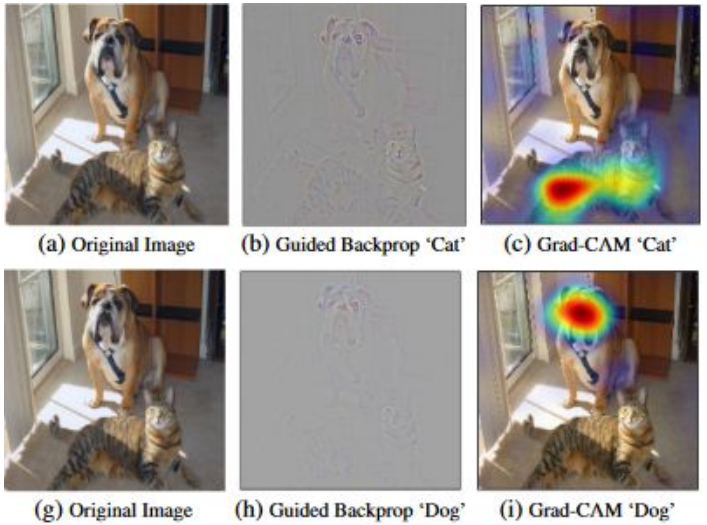
We define M_c as the class activation map for class c, where each spatial element is given by

$$M_c(x, y) = \sum_k w_k^c f_k(x, y). \quad (2)$$

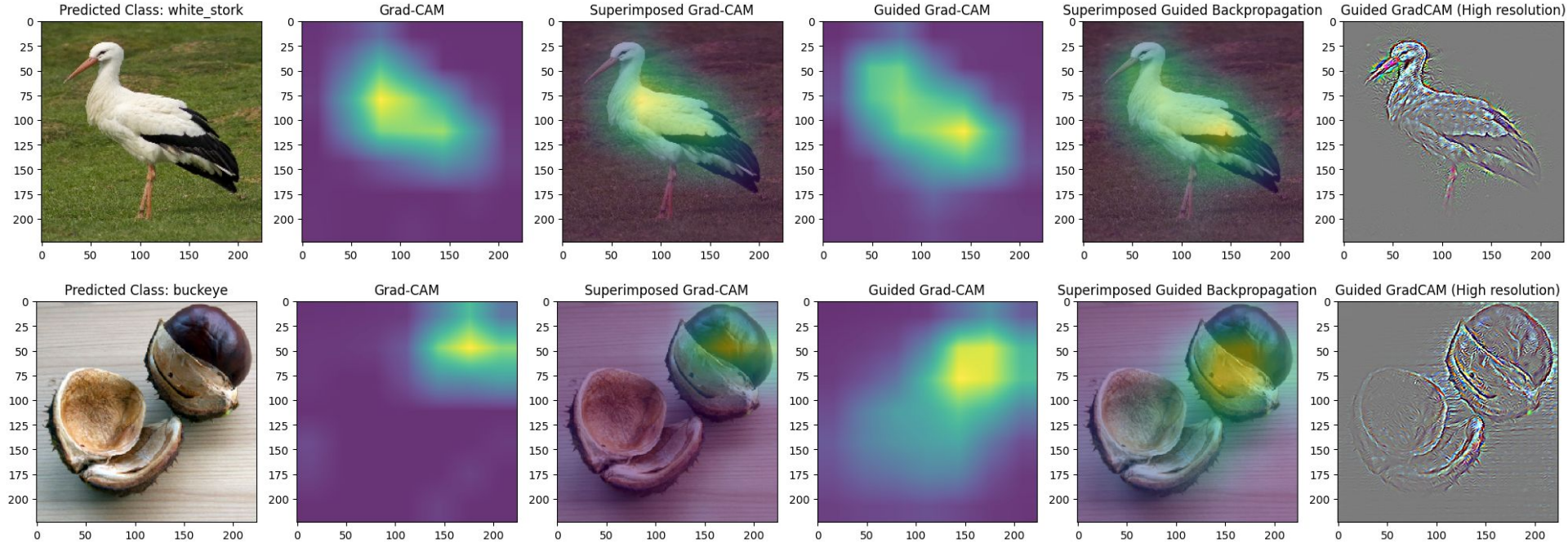
Thus, $S_c = \sum_{x,y} M_c(x, y)$, and hence $M_c(x, y)$ directly indicates the importance of the activation at spatial grid (x, y) leading to the classification of an image to class c.

Grad-CAM (Gradient-weighted Class Activation Mapping)

- **Objective:** Grad-CAM aims to provide insight into the regions of an input image that contribute the most to a neural network's decision when making a particular class prediction.
- **Method:** It combines the gradients of the class score with respect to the feature maps of the last convolutional layer.
- **Heatmap Visualization**
- **Applications:** Grad-CAM is widely used in interpretability and model debugging. It provides insights into why a neural network makes certain decisions.



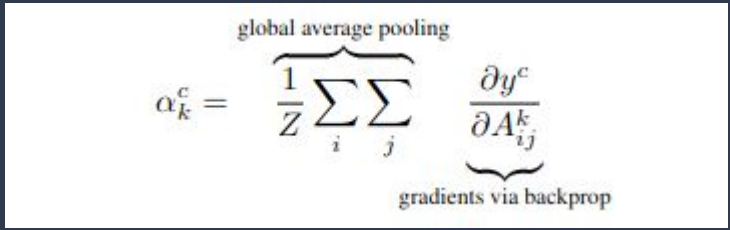
Selvaraju, R. R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.



Left to right: Original Image along with its predicted output by ResNet101 model. GradCAM heatmap, superimposition of GradCAM heatmap on original image, guided GradCAM with relu backpropagation heatmap, superimposition of guided GradCAM on original image and high resolution Guided GradCAM of the original image.

Grad-CAM

In order to obtain the class-discriminative localization map Grad-CAM, we first compute the gradient of the score for class c. These gradients flowing back are global-average-pooled over the width and height dimensions to obtain the neuron importance weights.



During computation of α^c_k while backpropagating gradients with respect to activations, the exact computation amounts to successive matrix products of the weight matrices and the gradient with respect to activation functions till the final convolution layer that the gradients are being propagated to. Hence, this weight α^c_k represents a partial linearization of the deep network downstream from A, and captures the 'importance' of feature map k for a target class c. We perform a weighted combination of forward activation maps, and follow it by a ReLU to obtain :

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Future Directions

- How explainability can help uncover decisions taken by CNN in regards to deepfakes.
- Comparative Study over various CAM models using various benchmark datasets.
- Gather more information to increase understanding of Explainable AI in domain of image forensics.

Thank You