Defense Against Adversarial Attacks using Convolutional Neural Network

Hao Wan
Volgenau School of Engineering ECE
George Mason University
Fairfax, VA USA
hwan5@gmu.edu

***Abstract*—Convolutional Neural Networks (CNNs) [1] have demonstrated remarkable performance in various recognition tasks, yet their vulnerability to adversarial attacks [2] remains a critical challenge. These attacks introduce small, imperceptible perturbations to input data, leading to misclassification and posing significant risks in safety-critical domains such as autonomous driving and facial recognition. To address this issue, I propose a Convolutional Self-Attention Auto-Encoder (CSAAE), a novel architecture that combines the strengths of convolutional layers and self-attention mechanisms. The CSAAE enhances model robustness by capturing both local and global features during image reconstruction, effectively mitigating adversarial perturbations. I also evaluate my approach against Fast Gradient Sign Method (FGSM) attacks on three benchmark datasets: MNIST, Fashion-MNIST, and Animal Faces. The results demonstrate that the CSAAE significantly outperforms traditional convolutional auto-encoders in reconstruction quality and adversarial defense. Furthermore, we deploy this method in an Android application, showcasing its practicality for real-world, on-device adversarial defense.**

Keywords—Neural Networks, Auto-Encoders, Adversarial Attacks, Transformer, Self-attention

I. Introduction

The rapid adoption of deep learning in real-world applications has exposed a critical vulnerability: adversarial attacks. These attacks involve subtle, often imperceptible modifications to input data that can drastically alter a model's predictions. Such vulnerabilities are particularly concerning in domains where safety and reliability are paramount, such as autonomous vehicles and biometric authentication systems.

To counter these threats, we introduce the Convolutional Self-Attention Auto-Encoder (CSAAE), a hybrid architecture designed to reconstruct clean images from adversarially perturbed inputs. By integrating convolutional operations with self-attention mechanisms, the CSAAE leverages both local feature extraction and global contextual understanding, enabling it to restore adversarial examples to their original classifiable forms.

This work makes three key contributions:

1: Architecture: We propose a U-shaped auto-encoder with self-attention layers, enhancing feature coherence across the entire image.

2: Robustness: We demonstrate the CSAAE's superior performance over traditional convolutional auto-encoders (CAEs), particularly on complex datasets.

3: Practical Deployment: We implement the CSAAE in an Android application which is convenient for users, validating its feasibility for real-time adversarial defense.

II. RELATED WORKS

The field of adversarial machine learning has evolved along two parallel tracks: the development of increasingly sophisticated attack methods and corresponding defense strategies. Seminal work by Szegedy et al. [3] and Goodfellow et al. [4] first revealed the susceptibility of deep neural networks (DNNs) to carefully crafted perturbations, establishing the foundation for modern adversarial research. Among attack methodologies, gradient-based techniques have proven particularly effective, with the Fast Gradient Sign Method (FGSM) and its iterative extension Projected Gradient Descent (PGD) emerging as standard benchmarks for evaluating model robustness. Subsequent taxonomy studies have further classified attacks along dimensions including targeted/untargeted objectives and white-box/black-box threat models.

Defense strategies against adversarial attacks generally fall into three categories:

1: Architectural Robustness: Methods that enhance model intrinsic resilience through techniques like adversarial training or Lipschitz-constrained layers. These modify the learning objective or network structure to improve resistance to perturbations.

2: Input Transformation: Approaches that preprocess inputs to remove adversarial noise while preserving semantic content. Notable examples include: Defense-GAN [5]: Leverages generative adversarial networks to project adversarial examples onto the clean data manifold. And PuVAE [6]: Employs variational autoencoders for probabilistic purification of corrupted inputs.

3: Detection Mechanisms: Auxiliary systems that identify and filter adversarial samples prior to classification, such as feature squeezing or anomaly detection modules.

Building on these foundations, recent approaches have explored convolutional auto-encoders (CAEs) for defense. These models aim to denoise adversarial examples by reconstructing inputs in a way that preserves class-relevant features. In contrast to heavier generative models, CAEs offer low-latency inference while achieving competitive accuracy restoration under adversarial conditions, making them appealing for real-time or safety-critical applications.

III. Implementation

This project investigates the effectiveness of integrating convolutional self-attention mechanisms into auto-encoders to improve the robustness of deep learning models against adversarial attacks. The proposed framework is evaluated on multiple datasets using standard convolutional neural networks (CNNs) and auto-encoder architectures, with a focus on classification accuracy. We also create a Android application to help user using this function in real world.

A. *Dataset implementation*

The proposed methodology is implemented using three publicly available, pre-processed datasets: the MNIST database of handwritten digits, the Fashion-MNIST database of Zalando product images and the Animal Faces dataset of real animal face.

The MNIST dataset consists of 60,000 training and 10,000 test images of handwritten digits (0-9) in grayscale format. Each 28×28 pixel image is center-normalized and preprocessed from the original NIST samples to ensure consistency. As one of the most widely used benchmark datasets in machine learning, MNIST provides a standardized testbed for evaluating classification algorithms on simple visual patterns.

Fashion-MNIST was designed as a more challenging alternative to MNIST, Fashion-MNIST contains Zalando's article images with identical dimensions (28×28 grayscale) and dataset splits (60k/10k). This drop-in replacement maintains MNIST's format while introducing greater complexity through 10 categories of fashion products, making it better suited for evaluating modern computer vision techniques.

The Animal Faces dataset [7] comprises high-resolution animal face images (typically 224×224 or 128×128 pixels) across multiple species. All samples are center-cropped and standardized, with high-quality annotations for supervised learning. The dataset's increased spatial dimensions and richer semantic content compared to MNIST variants make it particularly valuable for testing model robustness on complex, real-world patterns.

B. *Adversarial attack implementation*

Fast Gradient Sign Method (FGSM) one of the most widely studied attack methods. FGSM generates adversarial samples by adding perturbations aligned with the gradient of the loss.

FGSM is a simple and fast method to generate adversarial examples—inputs slightly modified to fool a neural network. It works by adding a small perturbation in the direction of the gradient of the loss. FGSM attacks can be described by the following equation [8]:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

Where:

x is the original input.

x_{adv} is the adversarial input.

ε is the perturbation size.

$\nabla_x J(\theta, x, y)$ is the gradient of the loss function J.

sign( ) is a sign function.

C. *Defense methods*

CAEs [9] are specialized auto-encoders that use convolutional layers for encoding and decoding. The encoder compresses input images into a latent representation through convolutional and pooling layers, while the decoder reconstructs the original image using transposed convolutions. CAEs excel at capturing spatial hierarchies but often lack global contextual awareness, limiting their effectiveness against adversarial perturbations.

CAEs can capture spatial hierarchies in images and it's better than fully connected autoencoders. Also, it can help us to compress high-dimensional image data into a lower-dimensional latent space.

D. *Self-attention mechanism*

Self-attention [10] is a dynamic feature-weighting mechanism that enables models to contextually relate all elements of an input sequence. Originally introduced in the Transformer architecture, it has become fundamental across machine learning domains, including computer vision.

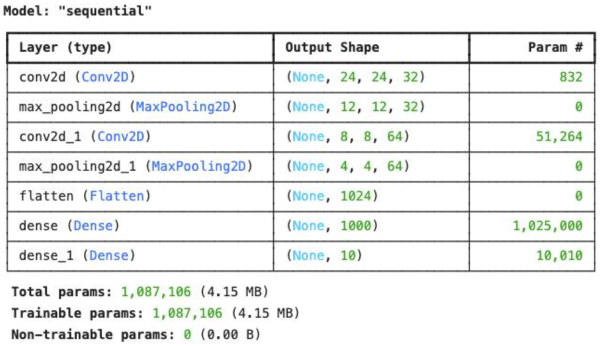
To illustrate with a simple example: imagine our input is a sequence of words: ["what", "sat", "sat", "upon", "hello", "mat"].

When processing the word "sat", self-attention asks a critical question: "Which other words in this sentence are most important for understanding 'cat'?" The model might determine that "sat" and "mat" provide more contextual relevance, and will therefore weigh these connections more heavily when creating the representation for "cat".

In our image defense context, this translates to identifying relationships between distant pixels. For instance, when reconstructing an adversarially perturbed digit or fashion item, self-attention helps our model maintain coherence across the entire image, not just locally. This global awareness is crucial for restoring the original patterns disrupted by adversarial noise.

E. CNN models

- Simple CNN: We created our own CNN model and used it on MNIST and Fashion-MNIST to avoid overfitting. The specific CNN architecture is showed in fig 1.
- VGG-16 [11]: It's architecture consists of 16 weight layers (13 convolutional layers with 3 × 3 filters, stride=1, and padding=1, plus 3 fully-connected layers) and 5 max-pooling layers using 2 × 2 windows with stride=2, processing 224 × 224 × 3 input images through progressively deeper feature representations before final classification via softmax output. This uniform design of small, repeated convolutional filters enables effective hierarchical feature learning while maintaining computational efficiency, with the systematic reduction of spatial dimensions through pooling layers and expansion of feature channels creating a powerful yet straightforward architecture that has become a widely-used benchmark for image classification and a versatile backbone for transfer learning in various computer vision tasks.



```
Model: "sequential"

┌──────────────────────────────────┬───────────────────────────┬────────────┐
│ Layer (type)                     │ Output Shape              │    Param # │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ conv2d (Conv2D)                  │ (None, 24, 24, 32)        │        832 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ max_pooling2d (MaxPooling2D)     │ (None, 12, 12, 32)        │          0 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ conv2d_1 (Conv2D)                │ (None, 8, 8, 64)          │     51,264 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ max_pooling2d_1 (MaxPooling2D)   │ (None, 4, 4, 64)          │          0 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ flatten (Flatten)                │ (None, 1024)              │          0 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ dense (Dense)                    │ (None, 1000)              │  1,025,000 │
├──────────────────────────────────┼───────────────────────────┼────────────┤
│ dense_1 (Dense)                  │ (None, 10)                │     10,010 │
└──────────────────────────────────┴───────────────────────────┴────────────┘

 Total params: 1,087,106 (4.15 MB)
 Trainable params: 1,087,106 (4.15 MB)
 Non-trainable params: 0 (0.00 B)
```

Fig. 1. CNN Architecture Details

F. Proposed Defense Architecture

We propose a novel Convolutional Self-Attention Auto-Encoder (CSAAE) that enhances traditional convolutional auto-encoders through strategic integration of self-attention mechanisms. As illustrated in Figure 2, the architecture employs a U-shaped design with the following key components:

Encoder Pathway: Processes adversarial inputs through multiple convolutional layers with GELU activations, progressively extracting hierarchical features while preserving spatial relationships through skip connections. The distinctive innovation lies in our integration of self-attention modules (visualized as yellow arrows in Fig. 2), which establish long-range dependencies between feature maps, enabling the model to maintain global contextual awareness during spatial downsampling.

Latent Space: The compressed representation is flattened into a 1×288 dimensional vector, where we apply controlled Gaussian noise injection. This critical design choice enhances robustness by training the model to reconstruct clean images from perturbed latent representations - effectively turning the adversarial attack strategy into a defense mechanism through noise-aware training.

Decoder Pathway: Mirrors the encoder structure using transposed convolutions for progressive upsampling. The decoder incorporates self-attention at its initial layer to preserve the global context established during encoding, ensuring coherent reconstruction of image-wide features disrupted by adversarial perturbations.

The complete system outputs reconstructed images that are subsequently classified by a pre-trained VGG-16 network, combining our model's denoising capabilities with established classification power.

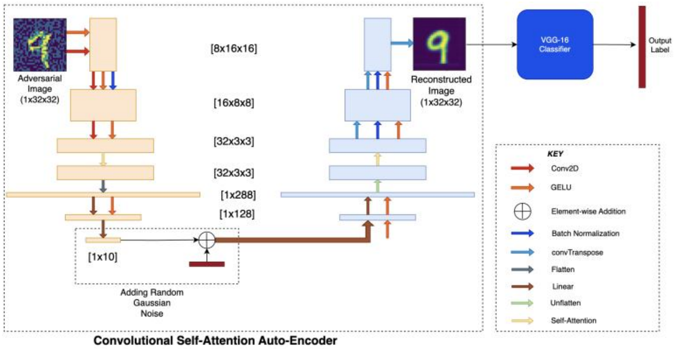


Fig. 2. Proposed Defense framework with a U-shaped Convolutional Self-attention Auto-Encoder and a pre-trained VGG-16 Classifier

Our Convolutional Self-Attention Auto-Encoder (CSAAE) processes adversarial inputs through its U-shaped architecture to output a reconstructed 28×28 image that effectively removes perturbations while preserving the original image's essential features, with this purified output then classified by a pre-trained VGG-16 network. We employ Gaussian Error Linear Units (GELUs) throughout the architecture due to their superior properties compared to ReLUs: their smooth, differentiable gradients enable more stable backpropagation; their non-zero outputs for negative inputs prevent neuron deactivation; and their probabilistic activation better models hierarchical features in deep networks - all critical advantages for our defense-focused model that must maintain robust gradient flow and feature diversity to successfully reconstruct images from adversarial examples.

G. Android User Interface

A focus of the application was to provide a clean and simple user interface and experience. We designed three functions within the app including image prediction, add adversarial attack and denoise image.

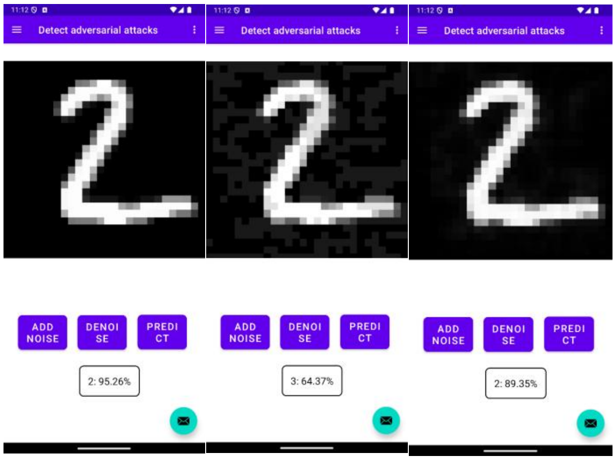


Fig. 3. Android app function show. Left one shows make prediction on the original image, middle one show the accuracy on adversarial image, the right one shows the prediction on denoise image

My Android application [12] UI including one slide window, three buttons and a text box. Slide window is used to show the image we will used. For three buttons, each one will call a function. Add noise will add FGSM attack on the current image. Denoise will used convoluional self-attention auto-encoder to defense the noise image. Predict is used to show the outcome and accuracy of current image.

H. FGSM algorithm

I wrote an algorithm to combine our CNN model with single FGSM and iterative FGSM attacks. It can depends on the way I put in to run the related codes automatically. Also, I chose to divided the whole formula into several parts to get the middle variables which helps us to check whether my code is right or not.

Algorithm 1: CNN with FGSM

```
Input: CNN Model (M), Original Sample (X, y), Perturbation
    Strength (ε), Step Size (a), Number of Iterations (N)
Initialize:
$\nabla = 0$
$X' = X$ + small random noise
if adversarial type is single FGSM
    Output = M(X)
    L = Loss(Output, y)
    $\nabla = \nabla_x L(M(X), y)$
    $X' = X + \epsilon \cdot \text{sign}(\nabla)$
else if adversarial type is iterative FGSM
    for i = 0 to N do
    Output = M($X'$)
    L = Loss(Output, y)
    $\nabla = \nabla_x L(M(X), y)$
    $X' = X' + a \cdot \text{sign}(\nabla)$
    Project $X'$ back into ε-ball around X
end for
Return $X'$
```

IV. EVALUATION

The main goal of this project is come up with a solution to defense the adversarial attack. I did several experiments including test the accuracy under different levels of adversarial attack, training the model with defence method CAE on MNIST and Fashion-MNIST dataset and use a pre-trained CNN model VGG-16 with CSAAE on the animal face dataset.

A. Computational platform and Software Environment

All experiments were conducted on an Apple MacBook with the following hardware and software configuration:

- Model: Apple MacBook (M1, Apple Silicon)
- CPU: Apple M1 chip (8-core: 4 performance cores + 4 efficiency cores)
- RAM: 16 GB unified memory
- Operating System: macOS Ventura
- Coding Environment: Jupyter Notebook
- Machine Learning Framework: TensorFlow (version 2.19.0)
- Python Version: Python 3.1.7

B. The Performance under different level of Adversarial Attacks

We tested our convolutional auto-encoder combined with simple CNN under different levels of adversarial attacks to show why adversarial attacks are dangerous for deep learning model.

We can see the data showed in table 1 and 2, with different levels of FGSM attacks, the deep learning model can be influenced deeply which disturb our model's function.

ε vs Accuracy for FGSM attack		
Epsilon	***Accuracy (w/o defense)***	
	MNIST	***Fashion-MNIST***
0.00	0.9919	0.9257
0.01	0.9232	0.4638
0.02	0.8457	0.3854
0.04	0.7529	0.2854
0.06	0.6126	0.2487
0.08	0.4257	0.1756
0.10	0.2532	0.1648
0.15	0.1723	0.0844
0.20	0.0739	0.0369
0.25	0.0721	0.0399

Table. 1. Accuracy for different level of FGSM attack.

C. Evaluate on the MNIST and Fashion-MNIST dataset

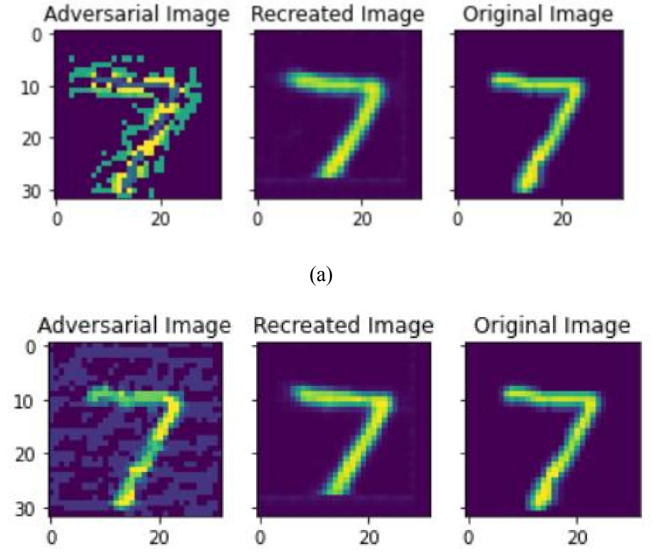
I combined the Convolutional Auto-Encoder (CSAAE) with simple CNN on the MNIST and Fashion-MNIST dataset. For each one, we tested the performance with/without defense under adversarial attacks FGSM where ε = 0.1.

Table 2 shows the accuracy of CAE on MNIST and Fashion-MNIST dataset. With defense method, our model can get a huge improvement compared with the accuracy without defense method.

Model	Attack	Accuracy(w/o defense)		Accuracy(with defense)	
		MNIST	*Fashion-MNIST*	*MNIST*	*Fashion-MNIST*
CAE	FGSM(ε=0.1)	0.2532	0.1648	0.9365	0.8429

Table. 2. Accuracy for CAE on MNIST and Fashion-MNIST.

For the (a) part, we added a single-step attack and the noise is large and rough, making the image more broken and distorted. For the (b) part, we used a muti-step, iterative attack which is more subtle and powerful, so the “7” is still more recognizable but better at fooling models.



(a)

(b)

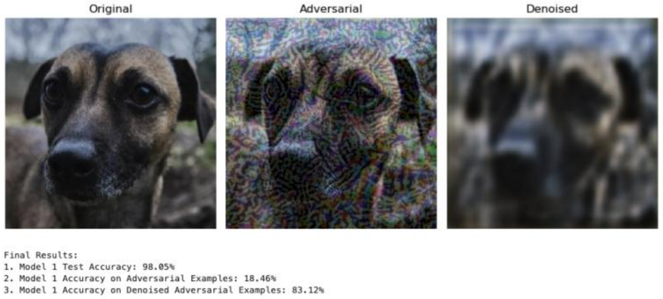
Fig. 4. Original Image, Adversarial Image and Recreated Image with (a) one-step FGSM attack (b) iterative FGSM attack

D. Evaluate on the Animal Face dataset

We combined our Convolutional Self-attention Auto-Encoder (CSAAE) with VGG-16 on the Animal Face dataset. and tested the performance with/without defense under adversarial attacks FGSM where ε = 0.1. Also Fig 5 is the instance of original, adversarial and denoise image on the animal face dataset.

Model	Attack	Accuracy(w/o defense)	Accuracy(with defense)
		Animal Face	*Animal Face*
CSAAE	FGSM(ε=0.1)	0.1846	0.8321

Table. 3. Accuracy for CSAAE on Animal Face.



```
Final Results:
1. Model 1 Test Accuracy: 98.05%
2. Model 1 Accuracy on Adversarial Examples: 18.46%
3. Model 1 Accuracy on Denoised Adversarial Examples: 83.12%
```

Fig. 5. Original, Adversarial Image and Denoise Image

E. Findings and Discussion

The experimental results across MNIST, Fashion-MNIST, and Animal Face datasets reveal that the proposed Convolutional Self-Attention Auto-Encoder (CSAAE) significantly improves model robustness against adversarial attacks. Under FGSM perturbations with ε = 0.1, models without defense exhibited drastic drops in classification accuracy — falling to 25.3% on MNIST, 16.4% on Fashion-MNIST, and 18.4% on Animal Faces. In contrast, incorporating CSAAE defense restored accuracy to 93.6%, 84.2%, and 83.2% respectively, demonstrating the model’ s strong denoising capabilities and resilience across varying data complexity.

These improvements validate our hypothesis that integrating self-attention with convolutional layers enhances both local feature extraction and global consistency during image reconstruction. In particular, self-attention contributed to better contextual awareness, allowing the model to recover semantically meaningful structures that adversarial perturbations had disrupted.

However, practical deployment introduced new challenges. The trained CSAAE model, at 384.6MB, exceeded the feasible size for on-device inference and could not be converted to TensorFlow Lite due to unsupported layers. To address this, we deployed the model on Google Cloud and used HTTP requests [13] to send input data and receive predictions. This solution, while effective, introduces latency and requires internet access, which limits fully offline usability.

Despite these constraints, our findings confirm the viability of CSAAE as an effective and adaptable defense mechanism, suitable for both academic evaluation and real-world integration through cloud-backed mobile applications.

V. Challenge

One of the key challenges I encountered was integrating my trained machine learning model into the Android application. The model’s size was approximately 384.6MB, making it too large for direct deployment on mobile devices. Additionally, attempts to convert the model into TensorFlow Lite (.tflite) format failed due to incompatible layers that were not supported by the TFLite converter. As a result, I explored an alternative solution by hosting the model on Google Cloud. This approach involved sending a POST request from the app with the user's input data, then receiving and parsing the JSON response containing the prediction result to display it in the app's UI. This cloud-based inference setup helped overcome both model size limitations and conversion issues.

VI. Conclusion

This project demonstrates that the Convolutional Self-Attention Auto-Encoder (CSAAE) is an effective and practical defense mechanism against adversarial attacks on deep learning models. By combining the strengths of convolutional feature extraction with self-attention's global context modeling, the CSAAE consistently restored corrupted inputs and significantly improved classification accuracy across MNIST, Fashion-MNIST, and Animal Face datasets. Experimental results showed substantial gains in robustness, especially in scenarios with strong adversarial perturbations.

Beyond algorithmic contributions, the successful deployment of this model in a functional Android application—despite conversion challenges and model size limitations—highlights its applicability in real-world scenarios. Hosting the model on Google Cloud and integrating it via RESTful APIs proved to be a feasible alternative for mobile deployment, balancing performance and practicality.

Future work will explore optimizing the model for on-device inference through pruning or quantization, extending defense capabilities to other attack types, and generalizing the framework to multi-modal or sequential data inputs. These enhancements will further strengthen the security and reliability of deep learning systems in safety-critical applications.

Addition Information

All code scripts have been made available at https://github.com/prediction-hao/ECE616-mobile-sys

References

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

[2] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. arXiv (Cornell University). http://arxiv.org/pdf/1706.06083.pdf

[3] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," Ieee Access, vol. 6, pp. 14410–14430, 2018.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.

[5] Samangouei, P., Kabkab, M., & Chellappa, R. (2018). Defense-GAN: Protecting classifiers against adversarial attacks using generative models. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1805.06605

[6] Lee, J., Kim, E., Kim, S., Lee, J., Yoon, S., & Park, S. (2019). Purifying Variational Autoencoder for Adversarial Defense. Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), 4364–4370. https://doi.org/10.24963/ijcai.2019/606

[7] https://www.kaggle.com/datasets/andrewmvd/animal-faces/data

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in International Conference on Learning Representations (ICLR), 2015

[9] Mandal, S. (2023). Defense Against Adversarial Attacks using Convolutional Auto-Encoders. arXiv preprint arXiv:2312.03520

[10] Zhao, H., Jia, J., & Koltun, V. (2020). Exploring Self-Attention for Image Recognition. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr42600.2020.01009M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[11] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. In International Conference on Learning Representations (ICLR).

[12] Android Developers. (n.d.). Build your first app. Retrieved from https://developer.android.com/training/basics/firstapp

[13] Fielding, R. T., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1: Semantics and Content (RFC 2616). Internet Engineering Task Force (IETF). https://doi.org/10.17487/RFC2616