

**Assignment 4 – Generative Adversarial Network**

Report

**HOZANA GUIMARAES REIS**

**Table of Contents**

[**Data Sample Images** 3](#_Toc162445781)

[**Data Preprocessing** 4](#_Toc162445782)

[**CNN for Generator Architecture** 5](#_Toc162445783)

[**CNN for Discriminator Architecture** 7](#_Toc162445784)

[**GAN training Step Function** 8](#_Toc162445785)

[**Train the Model** 8](#_Toc162445786)

[**Image prediction** 9](#_Toc162445787)

[**Conclusion** 9](#_Toc162445788)

# **Data Sample Images**

A computer code on a black background

Description automatically generatedThe dataset used is the data fashion\_mnist, below is a sample of the images that contains in the data followed by the code used. The data is stored ds1\_hozana and ds2\_hozana.

A collage of clothes and shoes

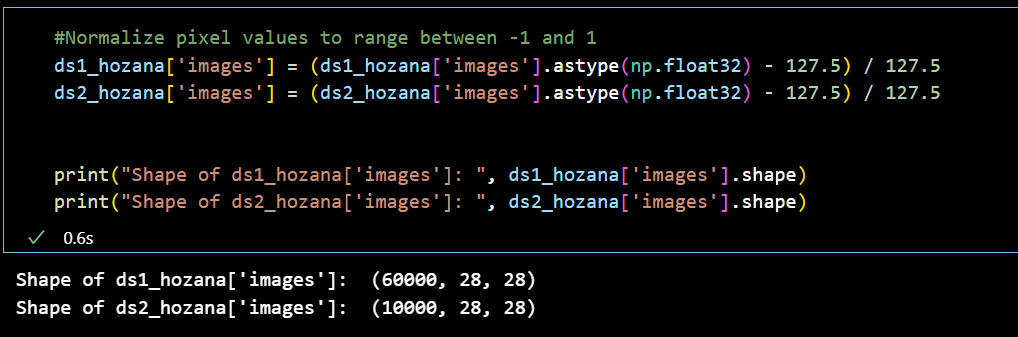
Description automatically generatedThe images contained in the dataset and the separated in classes as the images below:

A collage of different clothes

Description automatically generated

# **Data Preprocessing**

To ensure consistent and efficient model training, the pixel values of the images in both the training and testing datasets are normalized. This process involves scaling the pixel values to a range between -1 and 1. See below the code and shape output.



After Normalizing the data, I selected only pants images which is class 1 as per the previous class image. After selecting only pants the dataset has a shape if 7000,28,28. Check the code and Image demonstration.

A collage of different pants

Description automatically generatedA screen shot of a computer program

Description automatically generated

# **CNN for Generator Architecture**

In the figure below is the generator architecture summary and diagram.

A black and white rectangular sign

Description automatically generatedA screenshot of a computer program

Description automatically generated

# **CNN for Discriminator Architecture**

A black and white rectangular sign

Description automatically generatedIn the figure below is the discriminator architecture summary and diagram.

A screenshot of a computer screen

Description automatically generated

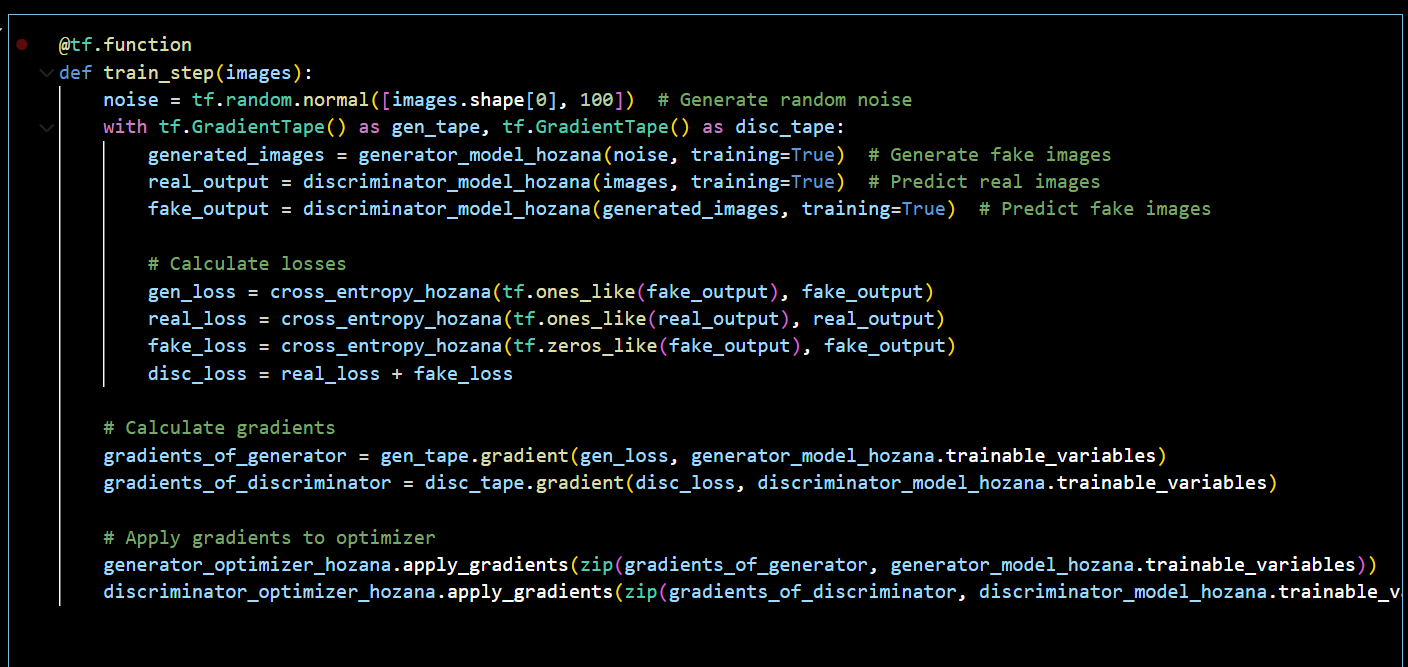
# **GAN training Step Function**

Basically what the function does is :

1. Generate Random Noise: Creates random noise to feed into the generator.
2. Forward Pass: Uses the noise to generate fake images with the generator, and predicts realness of both real and fake images with the discriminator.
3. Calculate Losses: Computes three types of losses:

* gen\_loss: Measures how well the generator fools the discriminator.
* real\_loss: Gauges how accurately the discriminator identifies real images.
* fake\_loss: Assesses the discriminator's ability to detect fake images.

1. Compute Gradients: Finds out how much each parameter should change to reduce the losses.



# **Train the Model**

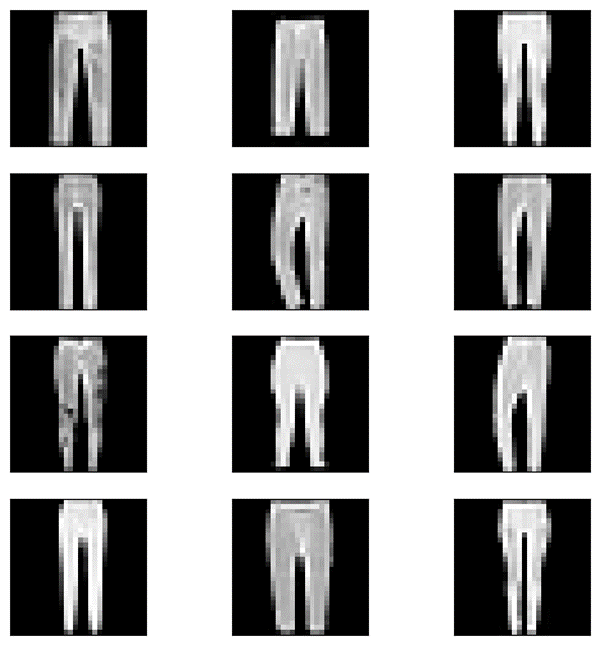
A screen shot of a computer

Description automatically generatedAfter the function I trained the model and displayed the epochs duration, see below the seconds taken for each epoch.

# **Image prediction**

Below is the image using the model, I also added the original image to compare.

A collage of black squares

Description automatically generated

Original

Predicted

# **Conclusion**

After looking at the predicted images next to the real ones, I noticed they're not exactly the same. The generated images mostly show clothes like pants or skirts, but they're not perfect. To make them better, I need to work on improving the GAN. This will helps create images that look more like the original ones.