

TABLE OF CONTENTS

APPROVAL PAGE	
PLAGIARISM STATEMENT	i
ACKNOWLEDGEMENTS	ii
CONTENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	viii
ABSTRACT	ix
I. Introduction	1
1.1 Research Background	1
1.2 Research Problem	2
1.3 Research Scope	2
1.4 Research Objective	3
1.5 Research Advantage	3
II. Literature Review	4
III. Basic Theory	10
3.1 Generative Adversarial Network	10
3.2 Conditional Generative Adversarial Networks Architecture	10
3.3 Loss Function	12
3.4 Discriminator Loss	13
3.5 Generator Loss	14
3.6 Combined Loss Function	14
3.7 Convolutional Neural Network	15
3.8 Convolutional Neural Network Architecture	16
3.8.1 Convolutional Layer	16
3.8.2 Pooling Layer	17
3.8.3 ReLu Correction Layer	19
3.8.4 Fully Connected Layer	19

IV. Research Methodology	20
4.1 Research Description	21
4.2 Image Dataset	21
4.3 Generative Adversarial Network	22
4.4 Optimized Loss Function for GAN	23
4.5 Convolutional Neural Network	24
4.6 Experimentation	25
4.7 Evaluation	26
4.8 Implementation	26
V. Implementation	28
5.1 Hardware and Software	28
5.2 Generative Adversarial Network Code Architecture	28
5.2.1 Data Preparation	28
5.2.2 Generator	30
5.2.3 Discriminator	32
5.2.4 GAN Loss	33
5.2.5 Model Initialization	33
5.2.6 The Main Model	34
5.2.7 Pretraining	37
5.3 Convolutional Neural Network Code Architecture	39
5.3.2 Creating The Model	39
5.3.2 Importing Detectron2 Utilities	40
5.3.3 Setting Up Bounding Boxes	41
5.3.4 Fusion Model Initialization	43
VI. Results	45
6.1 Images Used	45
6.2 GAN	45
6.3 CNN	46
6.4 Side by Side Comparison	47

6.5	Mean Absolute Error	50
VII.	Conclusion	54
	References	56

TABLE OF FIGURES

Figure 3.1 Visual Representation of GAN Architecture	12
Figure 3.2 Visual Representation of a Convolutional Layer	17
Figure 3.3 Visual Representation of Max Pooling and Average Pooling	18
Figure 4.1 Flowchart of the Research	20
Figure 4.2 Sample Images from COCO dataset	21
Figure 4.3 More Sample Images from COCO dataset	22
Figure 5.1 Importing COCO dataset into the code	28
Figure 5.2 Adjusting how the images will be used	29
Figure 5.3 Resizing images and making data loaders	30
Figure 5.4 Making the data loaders	30
Figure 5.5 The Generator	31
Figure 5.6 Continuation of The Generator	32
Figure 5.6 The Discriminator	32
Figure 5.7 GAN Loss	33
Figure 5.8 The Model Initialization	34
Figure 5.9 The Main Model	35
Figure 5.10 Continuation of the Main Model	36
Figure 5.11 Utility Function	36
Figure 5.12 Training Function	37
Figure 5.13 Making the pretrained generator	37
Figure 5.14 Pretraining the generator	38
Figure 5.15 Pretraining results after 20 epochs	39

Figure 5.16 Loading the pretrained model and weights	39
Figure 5.17 Downloading and unzipping the folder containing weights	40
Figure 5.18 Importing Detectron2 Utilities	41
Figure 5.19 Creating new path for bounding boxes prediction	42
Figure 5.20 Predicting bounding boxes in image folder	42
Figure 5.21 Setting Up Libraries and Options	43
Figure 5.22 Folder to save predictor color images	43
Figure 5.23 Initializing Fusion Model	44
Figure 5.24 Colorizing the images	44
Figure 6.1 Running the colorization process for 50 epochs	45
Figure 6.2 Results after 50 epochs	46
Figure 6.3 Selecting image to colorize from folder	46
Figure 6.4 Selected image colorized result	47
Figure 6.5 Colorized COCO images comparison	47
Figure 6.6 Another Colorized COCO images comparison	48
Figure 6.7 Colorized personal images comparison	49
Figure 6.8 Another colorized personal images comparison	49
Figure 6.9 Colorization on multiple shirts	50
Figure 6.10 Determining Absolute Error of GAN Image	51
Figure 6.11 Determining Absolute Error of CNN Image	51

TABLE OF TABLES

Table 2.1 Comparison of the related works	7
Table 6.1 MEAN of GAN and CNN	53