REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

COLORIZING GRAY LEVEL IMAGES WITH GENERATIVE ARTIFICIAL INTELLIGENCE

20011023 — Mehmet Alperen ÖLÇER
19011038 — Şevval BULBURU

COMPUTER PROJECT

Advisor
Prof. Dr. Mine Elif KARSLIGİL

June, 2024

ACKNOWLEDGEMENTS

Yildiz Technical University, founded in 1911 in Istanbul, Turkey, is a prominent government institution with 10 faculties, 2 institutes, and approximately 25,000 students. It is also regarded as one of the greatest in the country.

We would like to thank our instructor Mine Elif Karsligil for her significant assistance, continuous follow-up, and direction throughout our thesis on "Colorizing Gray Level Images with Generative Artificial Intelligence" Her knowledge of computer science, computer vision, algorithms, computer learning, artificial intelligence, image processing, pattern recognition, neural networks, technology and engineering was important in the project's success.

Mehmet Alperen ÖLÇER
Şevval BULBURU

TABLE OF CONTENTS

LIST OF SYMBOLS

Ł$*$	Lightness
log	Logarithm
μ	Micro
σ	Sigma
$\sum$	Sum of items

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Network
cGAN	Conditional Generative Adversarial Network
dB	Decibel
GAN	Generative Adversarial Network
LAB	L for lightness and a and b for the color-opponent dimensions of redness–greenness and blueness–yellowness
LPIPS	Learned Perceptual Image Patch Similarity
MSE	Mean Squared Error
Pix2Pix	Pixel to pixel
PSNR	Peak signal to noise ratio
RGB	Red-Blue-Green
SSIM	The structural similarity index measure
RAM	Random Access Memory
GB	Gigabayt

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

COLORIZING GRAY LEVEL IMAGES WITH GENERATIVE ARTIFICIAL INTELLIGENCE

Mehmet Alperen ÖLÇER
Şevval BULBURU

Department of Computer Engineering
Computer Project
Advisor: Prof. Dr. Mine Elif KARSLIGİL

Colorizing colorless photos and videos contributes to the preservation of historical and cultural heritage, making the past more accessible and understandable, and at the same time enriching aesthetic and artistic expression. Coloring gray-level photos is a challenging and not fully solved problem in the world of artificial intelligence. This problem, which is the subject of generative artificial intelligence, is a study topic with Pix2Pix approach. Instead of producing an output from nothing, an output is tried to be produced with a relevant input. In the studies conducted so far, different deep learning-based methods have been developed and the GAN architecture is based. In this study, cumulative studies leading to video coloring step by step were carried out. Different generator and discriminator models developed in the low-resolution dataset were compared and optimized, then adapted to high-resolution photos and continued. When photo coloring was successful, video coloring was successfully performed with information transfer between video frames and scene change control.

Keywords: Generative Artificial Intelligence, GAN, Pix2Pix, Image and Video Colorization, Information Transfer, Scene Change Control, frame extraction, CNN, U-Net, Deep Learning

ÖZET

GRİ SEVİYESİNDEKİ GÖRÜNTÜLERİ ÜRETKEN YAPAY ZEKA İLE RENKLENDİRME

Mehmet Alperen ÖLÇER
Şevval BULBURU

Bilgisayar Mühendisliği Bölümü
Computer Projesi
Danışman: Prof. Dr. Mine Elif KARSLIGİL

Renksiz fotoğraf ve videoları renklendirmek, tarihsel ve kültürel mirasın korunmasına katkıda bulunarak geçmişi daha erişilebilir ve anlaşılır kılar, aynı zamanda estetik ve sanatsal ifadeyi zenginleştirir. Yapay zeka dünyasında gri seviyesindeki fotoğrafları renklendirmek zorlayıcı ve tam olarak çözülememiş bir problemdir. Üretken yapay zekanın konusu olan bu problem Pix2Pix yaklaşım sınıfındadır. Hiçbir şeyden bir çıktı üretmek yerine, ilgili bir girdi ile bir çıktı üretilmeye çalışılır. Şimdiye kadar yapılan çalışmalarda farklı derin öğrenme tabanlı yöntemler geliştirilmiş olup, GAN mimarisi temel alınmıştır. Bu çalışmada adım adım video renklendirmeye giden kümülatif çalışmalar yürütülmüştür. Düşük çözünürlüklü veri setinde geliştirilen farklı üretken ve ayırt edici modeller karşılaştırılarak optimize edilmiş, ardından yüksek çözünürlüklü fotoğraflara uyarlanıp devam edilmiştir. Fotoğraf renklendirmede başarılı olunduğunda, video kareleri arası bilgi aktarımı ve sahne değişim kontrolü ile video renklendirme başarıyla gerçekleştirilmiştir.

Anahtar Kelimeler: Üretken Yapay Zeka, GAN, Pix2Pix, Fotoğraf ve Video Renklendirme, Bilgi Transferi, Sahne Değişim Kontrolü, Video Parçalama, CNN, U-Net, Derin Öğrenme

1
INTRODUCTION

Gray-scale images are memories that carry traces of the past. From old photographs to historical archives, these images are witnesses to history, evoking nostalgia and a desire for vibrancy. The process of colorizing gray-scale images builds a bridge between the past and the present, deepening our understanding and creating stronger ties to our past.

In the field of image processing, transforming gray-scale images to color remains an essential challenge. This challenge is analogous to translating a concept into multiple languages; a scene can be represented in a variety of visual forms, such as RGB images, gradient fields, edge maps, semantic label maps, and more. [1] Traditionally, these tasks have been undertaken by machines, and each of them was designed for a certain purpose. However, the introduction of deep convolutional neural networks has allowed for a more unified solution to the image prediction problem.

Deep convolutional neural networks have changed the landscape of computer vision by demonstrating remarkable success in learning complex features from visual data. Leveraging large datasets such as ImageNet and Pascal VOC, COCO, these networks have demonstrated the ability to scale and adapt to increasingly complex tasks. However, relying on large-scale labeled datasets poses challenges such as scalability, cost, and domain specificity. This has prompted the development of self-supervised learning methods, culminating in the creation of Generative Adversarial Networks.

The self-supervised learning process leverages unlabeled data, eliminating the requirement for manual labeling. Through different tasks such as image completion, super-resolution and image colorization, convolutional neural networks learn meaningful representations and bypass the limitations of human-prepared datasets. This method not only eliminates the amount of human effort required but also makes it easier to transfer gained features across domains. However, unlike traditional methods that rely on predefined loss functions, GANs dynamically learn a loss function and adapt to the data. This adaptability enables GANs to overcome the limitations of

traditional algorithms, opening new horizons in image manipulation and synthesis.

With their ability to learn and mimic complex patterns of real-world data, GANs demystify the colorization problem by filling gray-scale canvases with vibrant hues that resonate with human perception. With the subtle interplay between the generator and discriminator, GANs handle the fine details of color distribution and fill gray-scale images with realistic colors.

In this study, different GAN models were created to color gray level images using the CIFAR-10 dataset and the results were examined. Pix2pix translation was applied, and CNN was used together with cGAN to create the models. These models have been evaluated with various evaluation metrics to measure their success in colorizing images. Additionally, a comprehensive analysis was conducted to compare the performance of different GAN variants and determine which model was most effective in colorizing gray-scale images on the CIFAR-10 dataset. The results highlight the impressive performance and potential of cGAN and CNN in the field of image colorization.

Upon colorizing the gray images with the CIFAR–10 dataset, the colorization of gray level videos was examined. Video colorization is as important and challenging as coloring images. Its difference from coloring images is being able to detect scene transitions and use the same colors within the same scene. Considering this problem, a new method has been introduced by using GAN models used in coloring images. In this method, two different GAN models were created, working dependently on the previous frame and independently of the previous frame. A scene change detection algorithm was designed to determine which of the two models to use. Different datasets were created from various videos to evaluate the performance of the proposed model.

2
Preliminary Examination

Colorization of gray-scale images is an important task in image processing with various applications in fields such as historical image restoration, and entertainment. This paper covers two studies: the colorization of gray level images and the colorization of gray level videos. At the beginning of the project, the researches in the literature were examined and the requirements were determined. System analysis was performed and project planning was carried out. The design phase was analyzed under two different headings. The first one, the gray level image coloring research, describes a study in the field of gray-scale image coloring using Generative Competitive Networks on the CIFAR-10 dataset. Unlike traditional approaches, which mostly rely on user involvement or predefined requirements, machine learning models and image processing techniques have been used to automate the process of adding color to gray-scale images. The research aims to establish and test the effectiveness of different GAN models in adding realistic and visually appealing colors to monochrome images. In light of the results obtained from this research, GAN models are developed, a new dataset is created and the coloring of gray level videos is studied.

2.1 Related Works

In this section, studies on the colorization of gray level images and the colorization of gray level videos are presented. The methods for each topic have been extensively reviewed and the findings form the basis of the proposed proposal.

2.1.1 Gray Level Image Colorization

There have been significant improvements in the field of image colorization in recent years. Previous researches has shown that, GAN based methods provide more effective and realistic coloring outcomes compared to traditional methods in coloring gray level images. This chapter aims to highlight the place and importance of this work by providing a comprehensive summary of current research in the field of image

colorization.

In gray image coloring examinations, two automatic coloring algorithms have been discovered. Both of them were proposed in 14th European Conference, 2016 [2] [3]. In [2], the AlexNet model was trained using the ImageNet dataset. Later, the Pascal dataset was fine tuned to do classification, segmentation, and detection.AlexNet is the name of a convolutional neural network architecture, designed by Alex Krizhevsky in collaboration with Ilya Sutskever and Geoffrey Hinton [4]. AlexNet is an artificial neural network, specifically a convolutional neural network. It is intended to develop deep learning algorithms capable of performing effectively on enormous and complicated visual datasets. Although the outcomes were less successful than previous approaches, they were found to be more successful than other self-controlled models.Another finding from the experiments is that there is no difference in performance between training color images or gray-scale images. In [3], the VGG-16 model was trained on the Pascal dataset. VGG-16, a deep learning model created at the University of Oxford. VGG-16 is a convolutional neural network built on the ImageNet dataset that performs exceptionally well in classification tasks. When the results were evaluated, it was discovered that the segmentation task produced more successful outcomes than the other studies.

A further article on self-supervised representation learning was presented at the IEEE Conference in 2017 [5]. The writers used AlexNet as a foundation. The offered method proposes an architecture that creates two independent sub nets instead of the standard auto encoder by adding a split to the existing network. A network analyzes the gray-scale channel to predict color channels for image colorization. The other network predicts the gray-scale channel that provides the color channels. This method allows the model to extract features from the entire input signal. The model is pre-trained using the ImageNet and Places datasets. The PASCAL dataset was utilized to perform tasks including classification, segmentation, and detection after training process. The proposed method was tested on laboratory images and outperformed earlier self-supervised systems. Also, It overcomes some of the limitations of prior self-supervised approaches.

Another article published at the 2017 IEEE Conference in 2017 was examined. [1] cGAN was used in the presented method. cGAN is a technique based on GAN. According to this technique, the generator produces the image depending on a condition, instead of producing an image through a random vector. At the same time, Patch-Gan was suggested for the discriminator structure in the study. Patch-GAN is also known as 'Markovian Discriminator'. Accordingly, the image given as input is divided into N*N patches and the classification process is carried out. The model was

trained with the CityScapes dataset and then used for tasks such as segmentation and classification. By modifying the loss functions in the model, it was observed in the experiments that more successful results were obtained than GAN models.

One of the current research was published in 2022. [6] In the study, the model has been developed using cGAN. The COCO dataset has been used for training. The images employ LAB channels rather than RGB channels. In addition to the Min - Max loss function, Mean Absolute Error was employed. This allowed the generator to output more realistic visuals. After training, the model was fine-tuned with the Pascal dataset and applied to segmentation and multi-classification tasks. The results demonstrated that constructing models that would perform sub tasks using the weights of the original model, improved performance.

2.1.2 Gray Level Video Colorization

When the studies on video colorization are examined, it is understood that it is one of today's problems and is open to improvement. Especially after the development of deep learning models, different methods were tried using CNN. Following that, with the introduction of GAN models, GAN models for image colorization were developed and began to be applied to video colorization.

The methods used for video colorization can be divided into three categories: fully automatic [7] colorization, task-independent colorization [8], and sample-driven colorization [9]. The first studies rely on sample-driven colorization. Sample-driven colorization provides a method based on baselines and color ranges. In a 2017 study [9], a network structure known as Video Propagation Network was presented. It uses a two-component network structure to establish connections between the colors in the sample image and the pixels in the image to be colored and transfers the data to the new image. The DAVIS dataset was used to train the model. The results were measured with the PSNR technique. When the results were examined, it was seen that it produced a realistic coloring. Although the study reveals a suitable and innovative method for colorization, it produces output completely according to the given sample. In this case, the results produced by the model depend entirely on the correct sample given to the model.

With technological developments, task-independent methods have been developed to reduce human effort. In these methods, color information is output from the given data set instead of relying on specially given constraints. In 2018, Wei-Sheng Lai and colleagues developed a VGG network and presented both a task-independent model and a new method to reduce temporal differences between frames [8]. They measured

the performance with Perceptual similarity and Temporal stability and their results showed improvement compared to previous methods.

Fully automatic methods learn the transition from gray level images to color images. The 3-D CNN method is generally used to achieve this task. In the 2019 study [7], the model is organized using nearest neighbors in both bilateral and temporal domains. The model generates multiple colorized images and selects among them. The model was trained with DAVIS and Videvo datasets and measured with PSNR and LPIPS methods. The results of the model show better results compared to different methods, but the model requires more memory than its competitors.

Following the development of GAN models, the VCGAN model was presented in 2023, taking into account shortcomings such as temporal synchronization, algorithm, or human-dependent models [10]. VCGAN performs both image and video colorization through the same model. In this model, the basic GAN structure is created. The difference from GAN is that when the generator receives the last colored image, it becomes a video colorization model, while when it receives the current gray level image, it becomes an image colorization model. After model training and optimization, performance measurements were performed with PSNR and SSMI methods. Although the performance measurements do not differ much from other methods, the fact that the model can work for both image and video provides a solution to the shortcomings of previous studies such as memory and temporal synchronization.

3
System Analysis and Feasibility

In this section, system analysis and feasibility evaluations were made within the scope of the project. The feasibility of the project, technical infrastructure, resources and access to the project were examined.

The project's system analysis includes multiple crucial phases. First, expectations for utilization areas are a part of the requirements analysis and determination process. Second, specifying the technical elements of the re-colorization procedure is necessary to establish operational requirements. It's also essential to evaluate current re-colorization techniques and determine their shortcomings. Furthermore, comprehending the function of Generative Adversarial Network technology in image re-colorization requires examining its current state and future prospects. Finally, data analysis includes setting up the development environment by identifying the hardware and software requirements for GAN modeling, data processing, and model training. It also entails preparing gray-scale images for re-colorization and selecting appropriate color palettes for the re-colorization process. As previously stated, the project's objective is to create a model for coloring gray-scale photos using generative artificial intelligence which uses Generative Neural Network. In this regard, feasibility studies were conducted.

3.1 Feasibility

The feasibility studies were examined under the headings time, technical, legal, economic.

3.1.1 Time Feasibility

A three-month period was determined to be proper for project completion. The Waterfall approach was used to create the time feasibility aspect.

Gannt Schema for the project is given below at Figure 3.1.

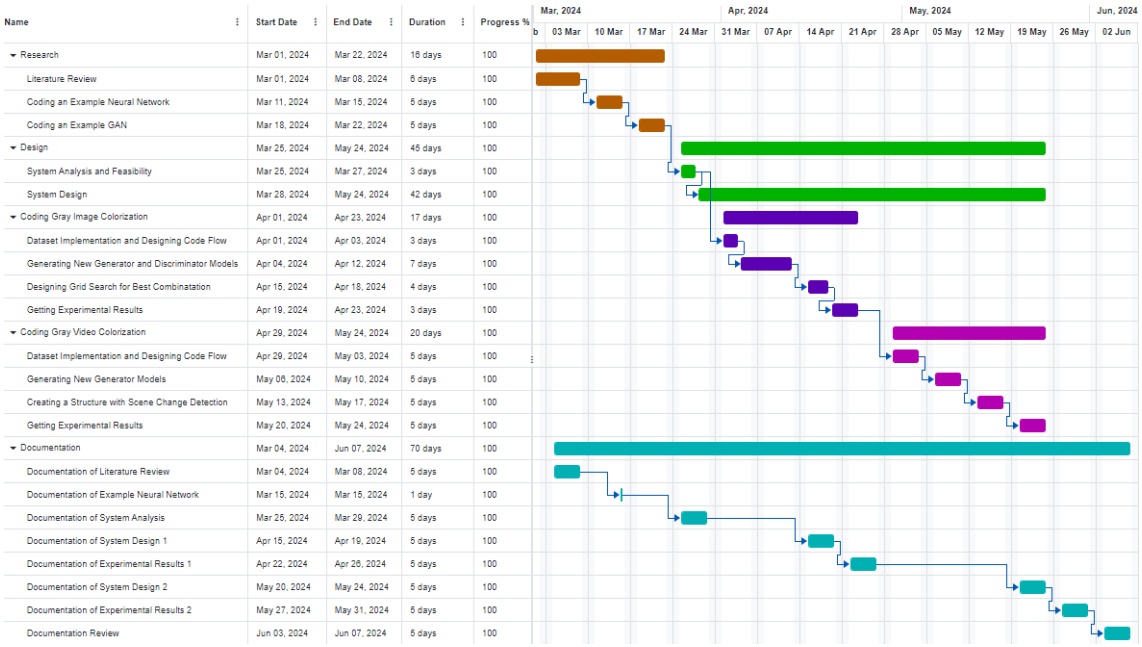

Figure 3.1 Gantt Schema for the Project

3.1.2 Technical Feasibility

To ensure the successful implementation of our project, technical requirements and resources have been carefully studied and analyzed, as detailed below.

3.1.2.1 Hardware Feasibility

During the development process, different types of models and datasets were examined; for example, CIFAR-10 dataset containing 60,000 images at 32x32 pixels resolution, 'Hababam Sınıfı' dataset containing 25,000 images at 128x128 pixels resolution, 'Documentary' dataset containing 150,000 images at 128x128 pixels resolution. Datasets other than Cifar-10 were created by the video frame extraction method. In the experiments, the following minimum specifications were sufficient for the most demanding training and usage case using 17 GB GPU memory and 25 GB RAM.

- Hard Drive: 250 GB - WDS250G3X0C
- Processor: AMD Ryzen Threadripper 2990WX 32-Core Processor
- Graphics: 24 GB - NVIDIA RTX A5000 Quadro
- RAM: 2 * KINGSTON 16 GB FURY BEAST RGB DDR4 3200MHZ

3.1.2.2 Software Feasibility

Python language was preferred for model training. The use of libraries such as TensorFlow, Matplotlib, scikit-image, NumPy, and Keras, as well as their user-friendly interfaces, validates their inclusion in the project. The Tensorflow library was used for the design of the model. Python version 3.10.12 and compatible library versions were selected.

3.1.3 Legal Feasibility

The project's goal is to offer an approach for converting gray-scale images and videos to three channel original images. The datasets used are CIFAR-10 and videos from YouTube, which are freely available for public use and do not require any rights purchases. In terms of software licensing, open-source resources were used, and because the software was developed internally, no license rights were violated.

3.1.4 Economic Feasibility

The cost of the hardware used is 136.000 TL. An employee works for 50.000 TL monthly. Two employees who work quarter-time for 4 days a week, are paid 2 * 50.000 * 0.8 * 0.25 = 20.000. The payment for two individuals for 3 months is 20.000 * 3 = 60.000. So for this project, the total salary is calculated as 60,000 TL. The total budget is 136.000 + 60.000 = 196.000 TL

Costs of each component in a computer are given below in Table 3.1.

Hardware - Cost Table	
Component	Cost
WDS250G3X0C	1.800 TL
AMD Ryzen Threadripper 2990WX 32-Core Processor	40.000 TL
NVIDIA RTX A5000 Quadro	90.000 TL
KINGSTON 16 GB FURY BEAST RGB DDR4 3200MHZ	2.100 TL
Total	136.000 TL

Table 3.1 Hardware - Cost Table

3.2 Elements of the System

The goal of this project is to create and implement an efficient image and video colorization algorithm that can bring visual appeal and new life to historic photographs, providing a fresh perspective on the past.

The Table 3.2 below provides an overview of the project's various resources, including hardware, software, human, and data resources.

Resources	
Resource	Element
Hardware Resources	Computer
Software Resources	Python3 - Tensorflow2
Human Resources	Mehmet Alperen Ölçer - Şevval Bulburu
Data Resources	CIFAR-10 Dataset and YouTube Videos

Table 3.2 Resources which used for the project

4
System Design

In today's technological age, numerous applications utilize the acquisition and processing of color images. In particular, synthesizing color images from gray-level images has attracted great interest in various fields such as art, medicine, security and industrial applications. Several techniques have been proposed to colorize gray-scale images. In line with the requirements analysis, firstly, various GAN models were created for the image colorization process and their performances were examined with the methods in the study. Then, for video colorization, scene change was detected and two different GAN models were created. An algorithmic structure was generated that allows both models to be used in video colorization. To measure the performance of the proposed method, two different datasets were created and the results were analyzed.

This section will provide a detailed analysis of the work performed and the system produced. The process of preparing the dataset and the design of the model, as well as how different models are created, will also be discussed.

4.1 Dataset Design

The image colorization model is trained using CIFAR-10 dataset images. CIFAR–10 is a subset of a dataset containing 80 million thumbnails. The dataset contains 60.000 images of 32x32 pixels. The pictures belong to 10 different classes, 6000 in each class. The CIFAR-10 dataset is a standard benchmark dataset frequently used in the fields of machine learning and image processing. This dataset is often used in machine learning algorithms to evaluate the performance of various object recognition and classification algorithms.

The CIFAR-10 dataset is downloaded using the TensorFlow library. The images that are obtained are in RGB format. The color space of RGB is not linear. In other words, the distances between colors are not perceptually meaningful. LAB color space ensures that the distances between colors are linear in a way that is more suitable for

human perception. Colors are intended to be more significant and consistent with this utility. In the LAB color space, color conversions are more stable, though. In light of these advantages, LAB color space was used during the conversion process rather than RGB color space in the training images. This preference allows for more accurate representation and rendering of colors, thus allowing for better analysis of colorful images. The dataset is given below in Figure 4.1

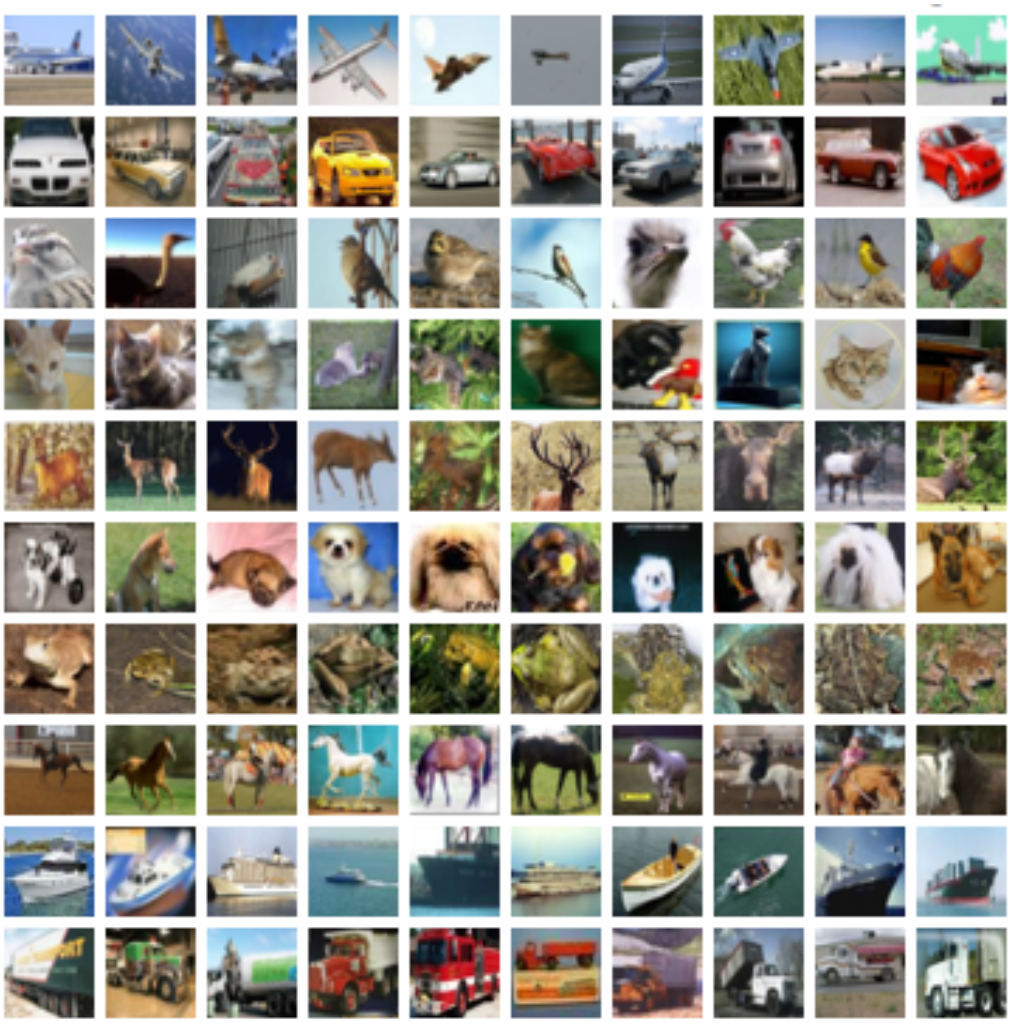

Figure 4.1 CIFAR-10 Dataset Images

For the video colorization model, it was decided to use 128x128 pixel images. Accordingly, the need to create a new dataset arose. As a result of the research, two different datasets consisting of YouTube videos were produced. In the first dataset, four movies from the 'Hababam Sınıfı' movie series with an average length of 90 minutes were selected. The second dataset consists of Swiss village videos and tropical island documentaries. For this dataset, 21 videos ranging in length from 20 minutes to 70 minutes were selected. One of the 'Hababam Sınıfı' movies was reserved for testing, others for training, and two documentary videos were reserved for testing, others for training. All videos were resized to 128x128 pixels and padding added to

top and bottom to avoid distorting the size when obtaining a square shape.

There are two different GAN models, independent and dependent, in the video colorization process. The video datasets determined for the training and testing of these models were divided into frames with the help of the openCV library. For the training of the independent GAN model, one frame was taken every 1.5 seconds in each video, while 3 frames per second were taken for the test of the model. For the Dependent GAN model, 3 frames per second were taken for training while 5 frames per second were taken for testing. As a result, while there are 25.000 images of 128x128 pixels in the 'Hababam Sınıfı' dataset, there are 150.000 images in the 'Documentary' dataset.

The model is first trained with the 'Hababam Sınıfı' dataset. The original coloring of the 'Hababam Sınıfı' dataset contains faded color sets. Considering this situation, the 'Documentary' dataset, which has more vivid colors, was created and the performance of the model was examined. Figure 4.2 shows the 'Hababam Sınıfı' dataset example, Figure 4.3 shows the 'Documentary' dataset example.

Figure 4.2 Hababam Sınıfı Dataset Images

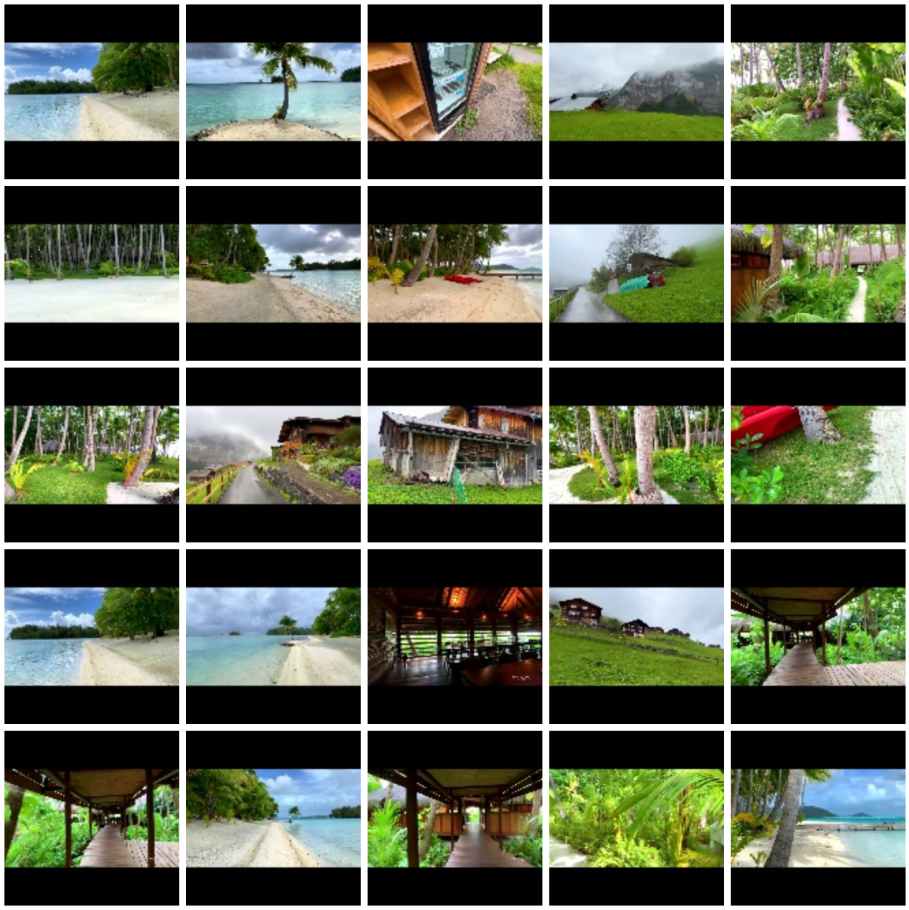

Figure 4.3 Documentary Dataset Images

4.2 Software Design

Colorization of gray-scale images is a significant challenge in computer vision that often presents as a complex and multidimensional problem. In this section, the GAN model designed to colorize gray-scale images, along with the Grid Search algorithm used to ensure the optimization process and improve the success performance, will be examined. In addition, two GAN models, independent and dependent, were generated for the video colorization process based on the Discriminator and Generator models selected based on the grid search optimization results. A scene change detection algorithm was created to determine which of the models to use. The procedures are presented in detail below.

4.2.1 Generative Adversarial Network

In 2014, Ian Goodfellow and associates introduced GAN, a deep learning model [11]. Deep learning models were often created via supervised learning and applied to classification tasks before the proposal of the GAN model. Yet, the endeavor of autonomously producing images from a dataset was revealed to be ineffective. The paper written by Ian Goodfellow introduced GAN to the world. GANs are a type of artificial neural network widely used in the field of deep learning. GANs do not require labeled data and can be trained using unlabeled data as internal data representation [12]. It performs particularly well in tasks like data augmentation, image creation, and image enhancement. GAN consists of two main components: Generator and Discriminator. These two elements are predicated on competing with one another to produce successful outcomes. The basis of this competition is game theory. Accordingly, the Generator tries to produce realistic data from randomly given noise. It attempts to approximate real data distributions in this process by rendering vector distributions more like real. Images that are initially unrealistic begin to become more realistic as the training process progresses. On the other hand, the discriminator attempts to differentiate between data generated by the generator and the real data. This can be described as a task classification problem. During training, the discriminator tries to give a high score for real data and a low score for fake data, while the generator tries to make fake data realistic.

Many different versions of GAN have been developed in later periods. The most suitable GAN type for colorizing gray-level images is cGAN. cGAN generators are designed conditionally. The generator produces data depending on this condition, instead of generating data from a random vector. Within the scope of the project, cGAN was designed by using gray-level images as input into the generator. cGAN's working principle is given as a blog diagram in Figure 4.4.

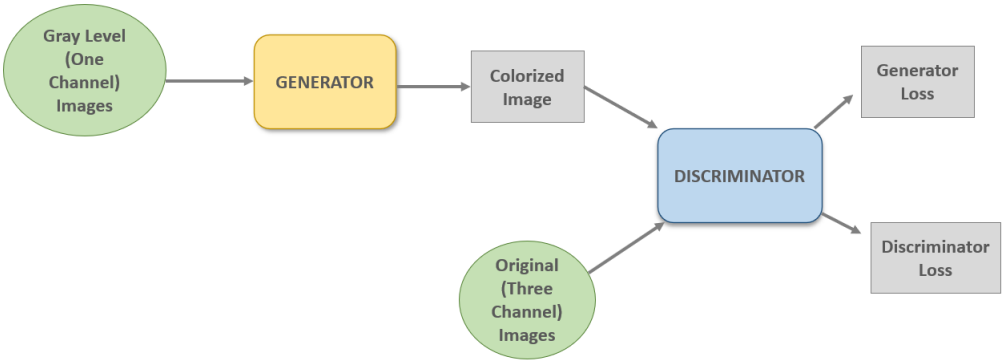

Figure 4.4 cGAN Blog Diagram

Generator and Discriminator work in opposition to each other, reinforcing each other. This development is achieved using Min-Max functions. The Generator attempts to produce realistic data based on the feedback the Discriminator provides during training. The generator tries to trick the discriminator by making the images it produces more realistic. Therefore, it aims to minimize the loss function during training. The opposite situation works for the discriminator. The discriminator tries to understand the difference between real data and generator-generated data. For this reason, while it gives high scores to real data, it gives low scores to fake images produced by the generator. In this case, the discriminator aims to maximize the loss function. Figure 4.5 represents standard Min-Max function.

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

Figure 4.5 Standard Min-Max Loss Function

The Min-Max function is divided into discriminator loss and generator loss based on the working principles between discriminator and generator. Their formulas are given below in Figure 4.6 and 4.7. The discriminator loss function is calculated using binary cross entropy to compare the difference between predicted and actual classes. While D(x) in the formula expresses the output of the discriminator, in other words, its probability, when real data is given, D(G(z)) refers to the output of the images created by the generator when given to the discriminator. Generator loss is described as a result of the discriminator's data classification. The stability of the loss functions is enhanced by expressing the acquired probability logarithmically.

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right]$$

Figure 4.6 Discriminator Loss Function

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log \left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

Figure 4.7 Generator Loss Function

The discriminator and generator must be in a balanced competitive environment. It is not ideal for one model to perform extremely well while the other does poorly. Usually, this balance has been obtained when designing the model and hyper-parameters.

4.2.2 Grid Search

Grid Search is a widely used parameter-tuning strategy in machine learning and model selection processes. This strategy aims to identify optimal parameter values to improve model performance.

Grid Search's working logic can be divided into three steps, the first step is to determine possible parameter values. These values can be chosen based on expert knowledge, experience, or data analysis results. In the second step, all combinations generated with these parameters are evaluated and a performance metric is calculated for each of them. In the last step, these metrics are subsequently organized according to a user-defined criterion or target metric. Finally, the best-performing parameter values are selected and the model is trained with these parameters.

Grid Search has several advantages, including a systematic and comprehensive search strategy, there is a high probability of finding optimum values in a given parameter space, and the results are repeatable and verifiable. Nevertheless, since it trains a separate model for all combinations, the computational cost increases and it is time-consuming when large parameter spaces or a large number of parameters are used. Therefore, it is not recommended when there are numerous possible combinations.

The use of Grid Search with GAN was carried out with the motivation of creating the most successful GAN model. Discriminator and generator in GAN can be considered as two separate models. These two models can be designed using different layers. As a consequence of this, using different models together and changing parameters can affect the success of the GAN model. For these reasons, two different discriminator models and two different generator models were designed. Combinations of these models with each other and other parameters used were obtained and their performances were measured using Grid Search. The experiments performed and the results obtained are examined in detail in Section 5.

4.2.3 Gray Level Video Colorization

Different methods have been presented so far for video colorization. The methods have been analyzed and a new method has been developed by us in the light of the findings obtained. The proposed method includes two different GAN models: Independent GAN and dependent GAN. Both models are created based on the results of the optimization process obtained by grid search. Then, a design is presented that enables the two GAN models to work according to the detection of scene changes.

The Independent GAN model has the same design as the GAN model used in the image colorization process. The generator consists of 5 up-sampling layers and 5 down-sampling layers. Discriminator has a CNN structure. The generator takes as input the L* channel in the LAB space of the image and generates the a and b channels. The task of the discriminator is to determine whether the given image is generated by the generator or not. For this purpose, it takes the image in the LAB space as input. The discriminator and generator work in cooperation to produce an image only from the light value of that image.

The dependent GAN model has the same discriminator structure as the independent GAN model. In the generator, the number of layers is the same but the number of input channels is 3. The first channel of these 3 channels is the L* channel of the frame to be colored, while the other two are the a and b channels of the previous colored frame. The generator establishes a coloring relationship between the current frame and the previous frame thanks to the a and b channels it receives as input. In this way, frames in the same scene are tried to be colored with the same color set.

Detection of scene changes is an important feature in video colorization. A design has been developed for scene change detection. Two images are compared for similarity and it is decided whether there is a scene change or not. Two methods are used in the comparison. In the first one, a similarity measure is obtained by comparing the histograms of the images. In the second, the SSIM score between the photos is calculated. If the histogram score is below 0.997 and the SSIM score is below 0.5, it is decided that there is a scene change.

Finally, a system that operates according to the detection of scene changes was created. In this system, the video to be colored is first divided into frames and each frame is transformed from RGB space to LAB space. Then, the first frame is given to the independent GAN model and the model performs colorization independently. The scene change detection algorithm determines which model will receive the next frames. If there is no scene change between the previous frame and the current frame, the dependent GAN model takes the color channels of the previous frame and the gray

level channel of the current frame and performs a colorization process depending on the previous frame. The block diagram of the generated design is shown below in Figure 4.8.

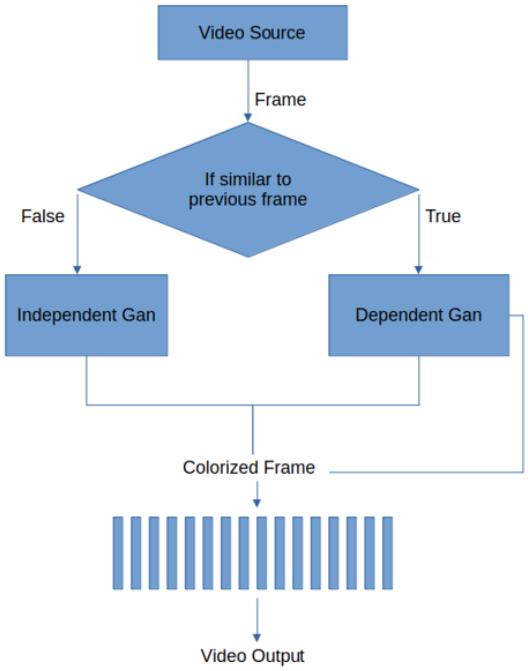

Figure 4.8 Video Colorization Block Diagram

4.3 Input - Output Design

As previously stated, the CIFAR-10 dataset was used in model training for image colorization. The dataset contains 60.000 32x32 pixels images. The images were converted from RGB to LAB space and used as ground truth for the discriminator, while their L* channels were utilized as generator inputs. The outcome of the generator has a and b channels. After the colorization process L*, a and b channels were combined and the color space was converted to RGB colorized 32x32 images. The Lightness channel of images, ground truths, and outputs are shown below respectively in Figure 4.9 .

In the video colorization process, the 'Hababam Sınıfı' and 'Documentary' datasets consisting of 128x128 pixels were used. The 'Hababam Sınıfı' dataset contains 25.000 images. The 'Documentary' dataset contains 150.000 images. The same image conversation and generation logic used in the CIFAR-10 dataset is also applied to these datasets. In Figure 4.10 and Figure 4.11 lightness channel of images, ground truths, and outputs are shown for both datasets.

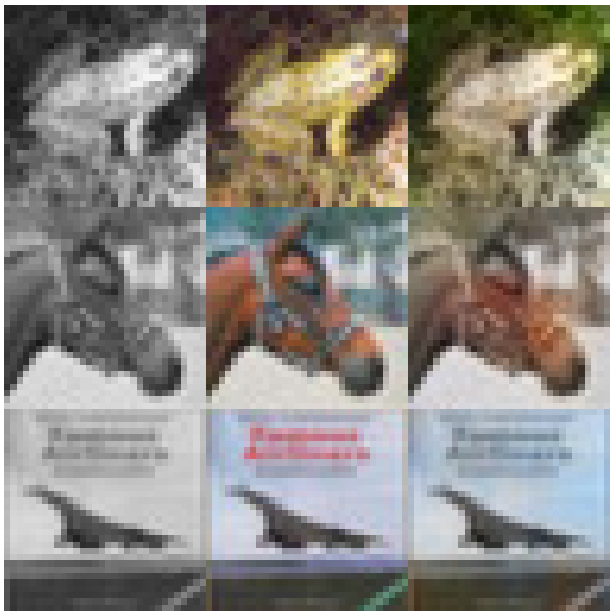

Figure 4.9 CIFAR-10 Dataset
Lightness Input - Ground Truth Input - Output from Model

Figure 4.10 Hababam Sınıfı Dataset
Lightness Input - Ground Truth Input - Output from Model

Figure 4.11 Documentary Dataset
Lightness Input - Ground Truth Input - Output from Model

5
Experimental Results

This section examines experiments conducted to colorize gray level images and gray level videos, as well as the results obtained.

The first development produced low-resolution imagery. These experiments were conducted using the Grid Search algorithm for image colorization. Different discriminator and generator model combinations were examined and their effectiveness was assessed. Several experiments were conducted to determine the effect of adjusting parameter values on model performance. The goal of the experiments is to identify the GAN with the best performance by observing how the combination of models together and changing the parameters affects the success of the GAN model.

After successfully re-coloring low-resolution images then developments moved on to higher resolution. The generator and discriminator are selected according to their performances, then modified and optimized for the next studies which are video recoloring.

5.1 Generated Models for Image Colorization - Low Resolution

As mentioned before, the CIFAR-10 dataset contains 60.000 images of 10.000 validation and 50.000 training for image colorization. Images are 32x32 pixels and the reason for this is lesser training times and better understanding and debugging while creating models. In the next developments, image sizes will be increased. For practical usage of the dataset, it was imported from 'tensorflow.keras.datasets' and downloaded.

GAN architecture includes two different neural networks which are generator and discriminator. In development, for testing and optimizing, two generator models and two discriminator models were designed. While creating models, decisions are mostly made with consideration not randomly and explained after every scheme.

5.1.1 Generator 1

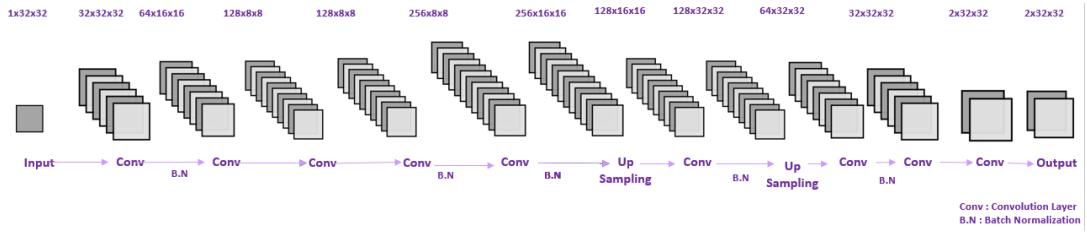

Figure 5.1 Architecture of Generator 1

As it can be observed from Figure 5.1 many layers are placed to make the model more complex for better learning. Since this might cause over-fitting also 5 times batch normalization applied. The 'Relu' activation function has been preferred as it has less running time due to simpler computational complexity with only max(0, x) operation. Adam optimizer' was preferred because it is well known for its faster convergence feature. The model takes gray-scale images and converts them to 2 channel images with CNN. This kind of transformation in GAN is called Pix2Pix, it is different from random noise taker generators.

5.1.2 Generator 2

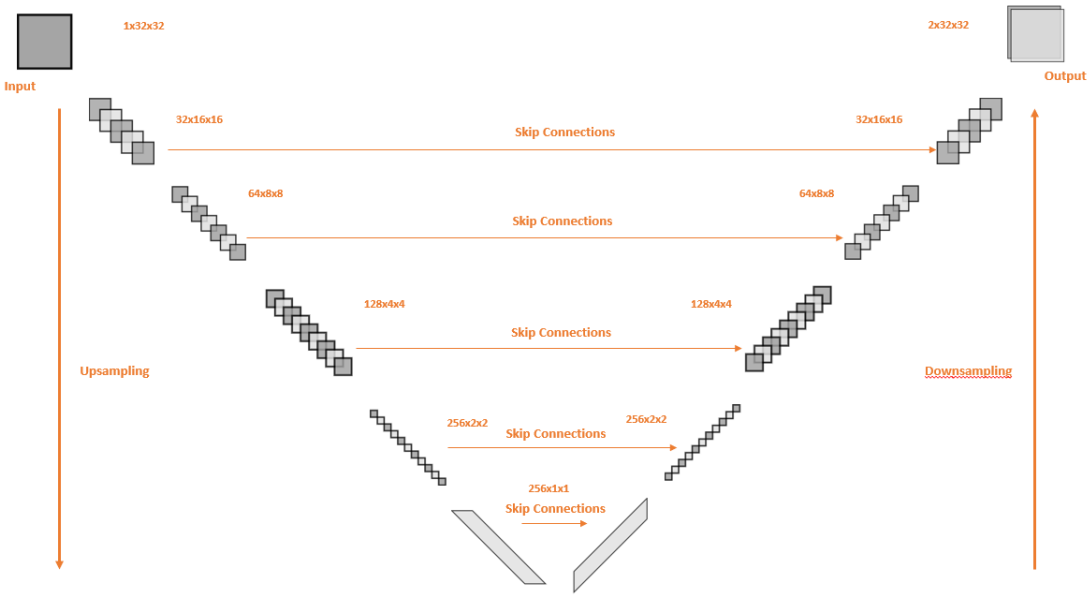

Figure 5.2 Architecture of Generator 2

Generator2 is created with layers and skip connections between layers. The model shown in Figure 5.2

5.1.2.1 Importance of Skip Connections in U-Net

In U-Net, skip connections connect the contracting (down-sampling) path to the expanding (up-sampling) path by copying and concatenating feature maps from the contracting path to the corresponding stage in the expansion path. These connections help preserve fine-grained details and allow high-resolution information to bypass sub-sampling processes, aiding in precise localization. They are vital for preserving spatial information during training and reducing the vanishing gradient problem, contributing to the effectiveness of U-Net in tasks such as image segmentation.

The most significant difference between these two generators is this one includes skip connections. With this feature, it becomes a U-net structure.

One major issue in image-to-image translation is solved by skip connections. Through these links, the information can move directly across matching layers, avoiding the bottleneck that encoder-decoder networks usually experience. This design decision enhances the preservation of significant details during the translation process by acknowledging the natural alignment between input and output structures. [1]

5.1.3 Discriminator 1

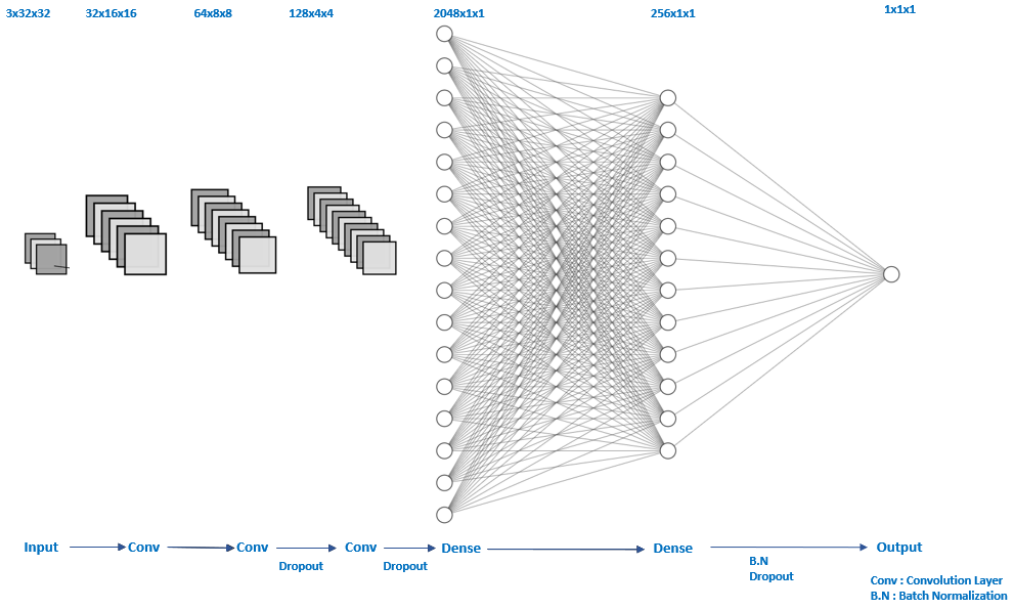

Figure 5.3 Architecture of Discriminator 1

Discriminator models input images are 3 channeled unlike generators and the output is 'True' or 'False'. If the input image is considered as it has original colors then it returns True. This network remains the simplest one when compared to others. The dropout method is applied 3 times and convolution connections use the 'Relu'

activation function but dense layers have 'Leaky Relu' to avoid the 'dead neuron' problem that might occur. 'Leaky Relu' solves this problem because its formula also allows for negative outcomes and its formula can be represented as f(x) = max(0.01*x, x). The model is shown in Figure 5.3

5.1.4 Discriminator 2

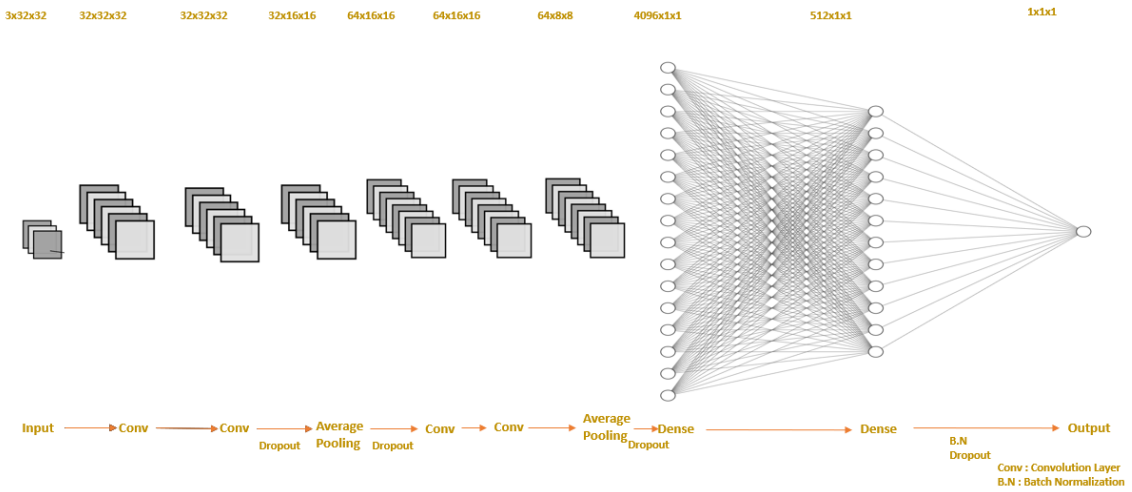

Figure 5.4 Architecture of Discriminator 2

This discriminator model is more complex than the first one for deeper learning. As can be observed from Figure 5.4, more convolution layers were added with 2 pooling layers. In CNNs pooling layers serve two main purposes: down-sampling feature maps to reduce computational complexity, and promoting translation invariance by aggregating local information, making the network less sensitive to small translations or distortions in the input. For our model average pooling method was chosen because while its strongest competitor max pooling method extracts more obvious characteristics like edges, it extracts features more smoothly.

5.1.5 Grid Search Experiments

Like most artificial intelligence projects a testing and optimizing model with different configurations were required. To overcome this issue as an automatic system a grid search approach was implemented. A code flow is written to test every combination of different parameters defined before runtime. In every training runtime over epochs, result metrics, test image results, and checkpoints are saved. Parameters used such as generator model, discriminator model, generator optimizer, discriminator optimizer, and batch size. With 2 generator models, 2 discriminator models, 1 generator optimizer, 1 discriminator optimizer, and 2 different batch sizes a configuration defined as Table 5.1

Table 5.1 Grid Search Space

Generator	generator1, generator2
Discriminator	discriminator1, discriminator2
Generator Optimizer	name:adam - learning-rate:2e-4, beta-1:0.5
Discriminator Optimizer	name:adam, learning-rate:2e-3, beta-1:0.7
Batch Size	128, 256

Each GAN model's run-times depending on the epochs are shown in Figure 5.5 as graph and the results obtained are shown in Figure 5.6.

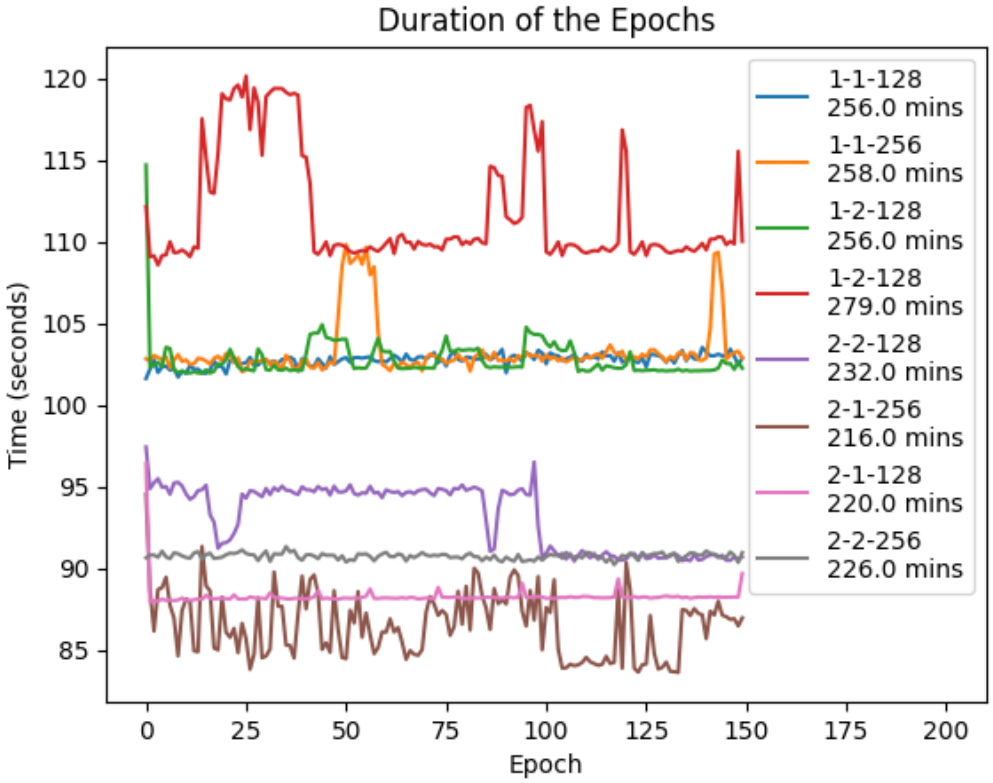

Figure 5.5 Duration of epochs at Grid Search, X-Y-Z-sum in name stands for Generator X, Discriminator Y, Batch Size Z

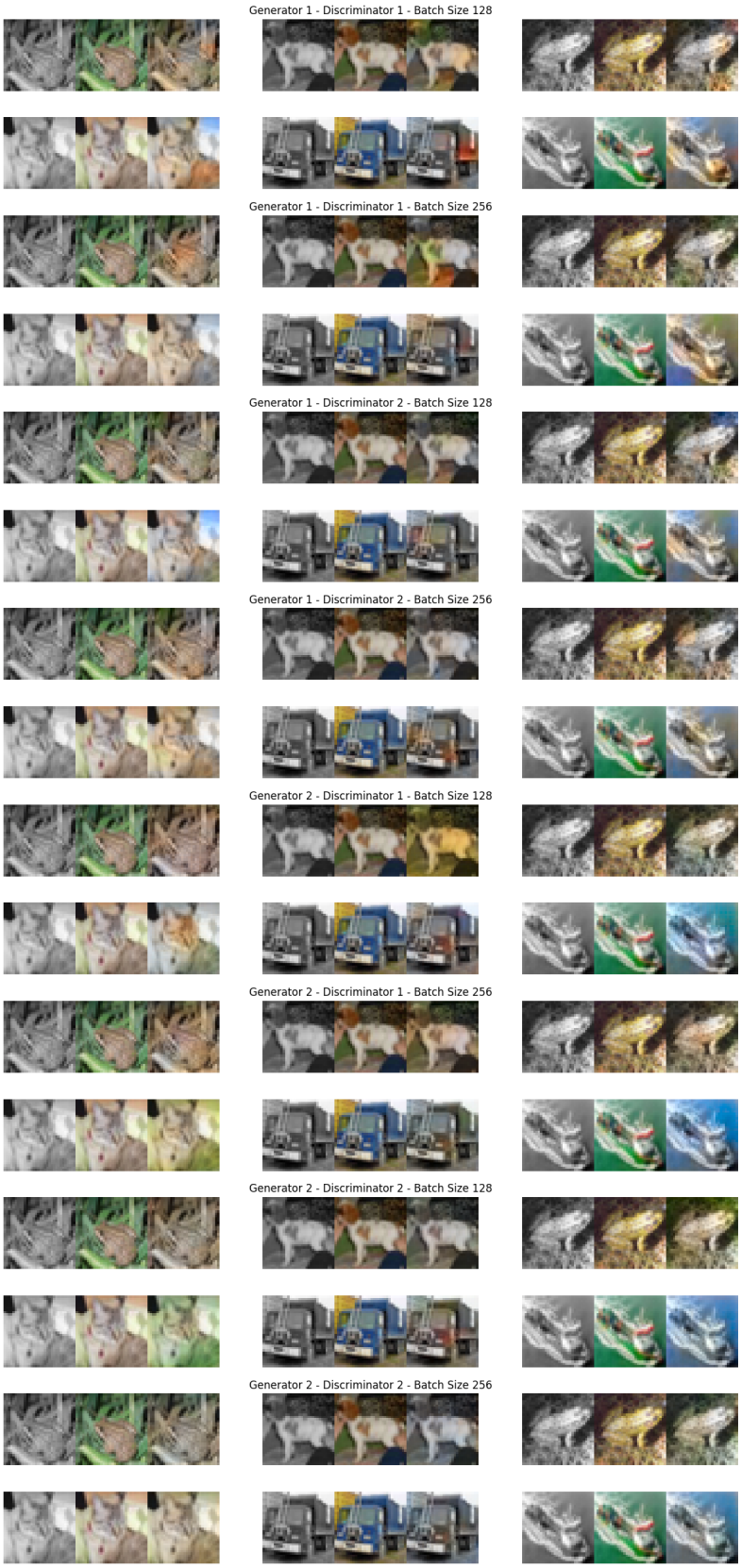

Figure 5.6 Left:Gray, Middle:Ground-Truth, Right:Output

Metrics used to measure the performance are the two most popular image reconstruction metrics: PSNR and SSIM.

PSNR measures the similarity and noise level of the original signal versus a copy of that signal. When used in GAN, the original signal corresponds to ground truth images, whereas the referred to as copy signal refers to images colored by the generator. PSNR is defined as the dB difference between signal and noise. A high PSNR value indicates higher quality, whereas a low PSNR value indicates increased information loss and lower quality. PSNR metric can be calculated using MSE [13]:

$$PSNR = 20\log_{10}(MAX_I) - 10\log_{10} MSE$$

SSIM is a metric used to evaluate the quality of a signal or image. Taking a different approach from PSNR, it takes into account the perception characteristics of the human visual system when measuring the similarity between two images. This allows detecting the loss of structural information in the image more precisely. The similarity score between two images is calculated using brightness, contrast, and structural similarities. Measurement results range between 0 - 1 and results closer to one, indicate that the two images are more similar.

MAXI - a max level of intensity, usually 255. SSIM metric is calculated on various windows of an image. The measure between two windows x and y y of common size N x N is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

$$with\ c_1\ and\ c_2 - two\ variables\ to\ stabilize\ the\ division\ with\ weak\ denominator. \tag{5.1}$$

$$\mu_x,\ \mu_y - average\ value\ for\ window. \tag{5.2}$$

$$\sigma^2{}_x,\ \sigma^2{}_y - variance\ value\ for\ window.\ \sigma_{xy} - covariance\ for\ windows\ x\ and\ y. \tag{5.3}$$

From the 4 GAN models, SSIM and PSNR scores of the generated images were measured. The performance of the measurements is shown in table 5.2. Analyzing the SSIM and PSNR results along with the Input-output images, it is observed that the model created with Generator 2 and Discriminator 2 gives better performance. In

the next stage, this GAN model will be used for both video colorization and image colorization.

Table 5.2 Metrics for each Combination, a:min, b:max, c:mean, X-Y-Z in name stands for Generator X, Discriminator Y, Batch Size Z

Name	SSIM a	SSIM b	SSIM c	PSNR a	PSNR b	PSNR c
1-1-128	0.80	0.99	0.91	14.80	34.37	22.90
1-1-256	0.81	0.99	0.92	16.00	36.40	23.65
1-2-128	0.81	0.99	0.92	14.68	34.72	22.91
1-2-256	0.80	0.99	0.92	15.25	33.20	23.55
2-1-128	0.82	0.97	0.92	16.61	29.37	22.95
2-1-256	0.82	0.97	0.92	14.86	27.80	22.62
2-2-128	0.81	0.99	0.92	15.34	35.15	23.30
2-2-256	0.81	0.99	0.93	15.04	35.02	24.07

5.2 Video Colorization - High Resolution

The grid search results we made are examined and it is seen that discriminator 2 and generator 2 are more successful. These two models are optimized and modified to run at higher resolutions.

When the model colors each frame while creating a video, the same object can be colored differently in consecutive frames. To prevent this, it is checked whether there is a scene change, and if there is no change, information is transferred from the previous colored frame to the current re-coloring process. The adaptation of this in the video creation structure is examined in detail in 4.2.3. We will call our previous frame-dependent coloring model as 'Dependent', and our independent model as 'Independent'. While the dataset we used for 128x128 pixel video coloring was examined in detail in 4.1, the discriminators and generators in these two models are examined below.

5.2.1 Generator - Independent and Dependent

Dependent and independent generator model structures are analyzed in below sections.

5.2.1.1 Independent

The better performance generator selected from the CIFAR-10 experiments called 'Generator 2', was modified and optimized. The layers and the structure are kept the same, only the data size is multiplied by 4 which means input, output, and hidden

layers data sizes increased. In conclusion, the generator becomes suitable to 128x128 pixel images. As a result of these changes, the structure was changed from 5.2 to 5.7.

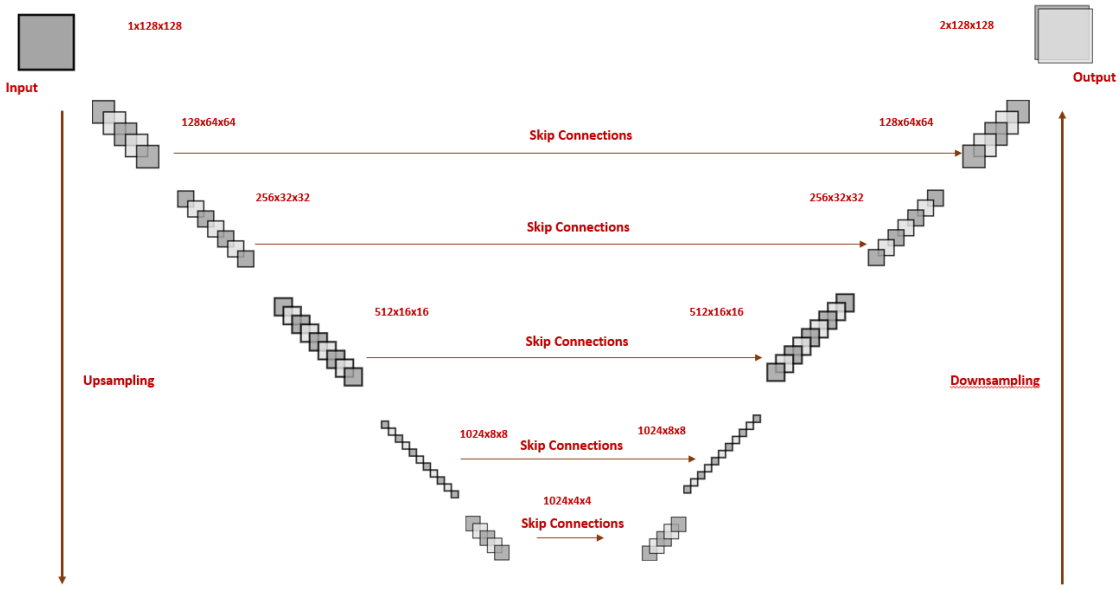

Figure 5.7 Architecture of Generator Independent

5.2.1.2 Dependent

The only difference between independent and dependent generators is the input channel size. In independent input image has one channel which is L* of LAB but in dependent input image has three channels accordingly L* of the current frame, a and b of the previous frame. It also gives output with 2 channels which are current a and b, like independent ones.

5.2.2 Discriminator - Independent and Dependent

The same discriminator evaluated from the CIFAR-10 experiment, called 'Discriminator 2', was used in both the Independent and Dependent models. About the changes, two additional convolution layers, a max pooling layer and an average pooling layer were added to meet the resolution requirements. Additionally, the number of convolution filters was increased to improve feature extraction. As a result of these changes, the structure was changed from 5.4 to 5.8.

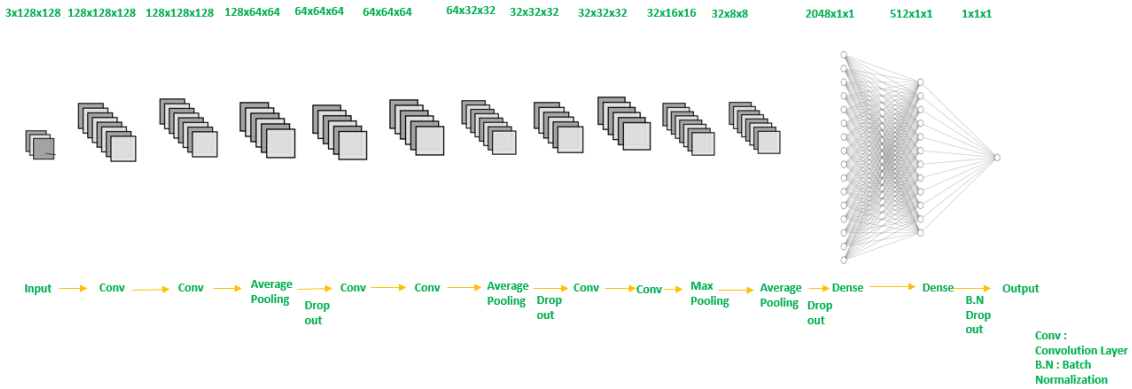

Figure 5.8 Architecture of Discriminator Independent

5.2.3 Scene Change Detection Algorithm Experiments

Four different techniques were tested for potential solution of scene change detection; ORB key-point extraction, Ffmpeg filters, histogram similarity, SSIM score.

- ORB Key-point extraction: ORB Key-point extraction is a technique that makes certain inferences by detecting key points in images. It combines the FAST and BRIEF algorithms. While the FAST algorithm detects key points, the BRIEF algorithm extracts features around the detected key points. When the ORB algorithm is used for a scene change, key-points are first detected for two frames. Then an analysis is performed by matching the key-points between the two frames. According to the analysis, it is decided whether there is a scene change or not. After some tests it has seen that this method is not suitable for low resolution images.

- Ffmpeg Filters: Ffmpeg is an open-source project for processing multimedia data on linux. It detects large transitions between frames by applying filters to the frames. In this way, scene transitions are detected. This feature was tested but unfortunately the accuracy of finding scene changes was around %50.

- Histogram Similarity: Detecting scene transitions by histogram similarity is done by analyzing the distribution of images in the frames. The gray-level histogram of each frame is extracted and the histogram differences of consecutive frames are calculated. Based on a threshold value, scene transitions are detected.

- SSIM Score: The SSIM score is a measurement method used to measure the similarities between two images. It tries to mimic human vision. In this method, SSIM scores are calculated for consecutive frames and scene transitions are detected according to a threshold value.

The tests were made on a video and the results were analyzed. As a result of the analysis, histogram similarity and SSIM scores calculating methods were hopeful results. To gain more accuracy, both of them were used in a condition mentioned in 4.2.3, and a well-working algorithm was created.

5.2.4 Results

The model created for the video colorization process was trained separately with the 'Hababam Sınıfı' dataset and the 'Documentary' dataset. The'Hababam Sınıfı' dataset contains more people and has a paler color set. The 'Documentary' dataset consists of Swiss villages and tropical islands in a natural environment. It also has a more vivid color set. Both datasets were trained and their performance was measured with test datasets. 5.9 shows 16 frames from successfully generated videos for both datasets. Additionally, the SSIM and PSNR scores of the resulting colorized frames are examined in 5.3.

The full versions of the colorized videos are linked in the Section A .

Table 5.3 Metrics for Hababam and Documentary Datasets, a:min, b:max, c:mean, X-Y-Z in name stands for Generator X, Discriminator Y, Batch Size Z

Name	SSIM a	SSIM b	SSIM c	PSNR a	PSNR b	PSNR c
Hababam	0.607	0.986	0.964	19.759	35.423	27.302
Documentary	0.759	0.986	0.953	15.203	33.235	25.356

Figure 5.9 Lightness Input - Ground Truth Input - Output from Model
Documentary - Hababam Datasets
Video re-coloring results for 16 frames

There are frames where the model fails as well as succeeds in colorization. 5.10 shows the unsuccessfully colored frames for both datasets.

Figure 5.10 Lightness Input - Ground Truth Input - Output from Model
Documentary - Hababam Datasets
Wrong Re-colorization Examples

6
Performance Analysis

6.1 Runtime Performance of Model and Algorithm

Since this project is not meant for live applications, its main goal does not require quick runtime performance. Nonetheless, both performance and speed were carefully taken into account during the development period. The following describes the runtime performance for 128x128 pixel images throughout the re-colorization process:

- Dependent model time: 0.0217 seconds
- Independent model time: 0.5777 seconds
- Scene change detection time: 0.0046 seconds

These measurements were taken from a 1000-frame recolorization test. The findings show that it takes 27 seconds in total to colorize 1000 frames. As a result, 27/1000 = 0.027 seconds, or 27 milliseconds, is the average time per frame and translates to a frame rate of 37 frames per second (FPS).

It is clear from looking at these computations and outcomes that using the dependent model greatly speeds up processing.

6.2 Performance in Training and Resource Usage

In most of the deep learning projects, the model evaluation part covers most of the work and study. Usually, even a small change in model training; affects the significant performance of the model, training duration, or resource usage. The duration of epochs in low-resolution dataset training was presented in Figure 5.5 and the records for dependent and independent models are shown in Figure 6.1.

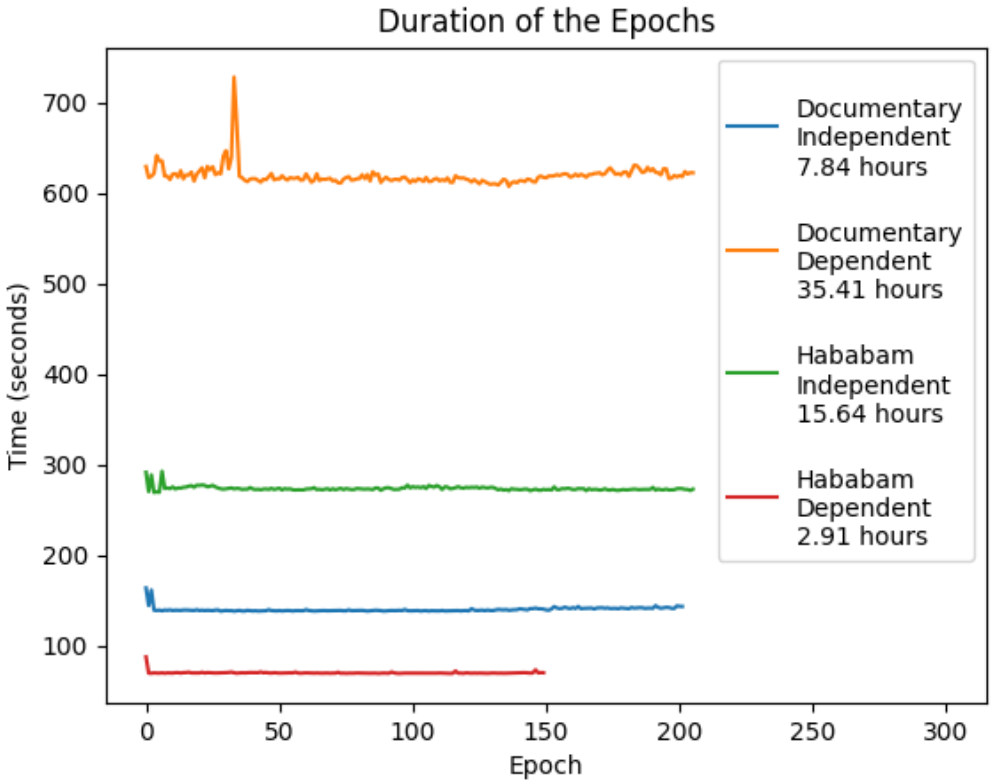

Figure 6.1 Duration of epochs at training

There were more than 10 training sessions while doing experiments and the resource requirements were different. Considering the biggest dataset and most complex model experiment the resource usage was the most and the record of that case was taken while training. The dataset was 150.000 128x128 pixel images and the model was dependent on GAN. The usage can be seen in Figure 6.2 which is recorded at the last epoch of the training session. The GPU temperature, GPU power usage, GPU memory usage, GPU utilization, CPU core activity, and RAM usage can be obtained from the record.

Figure 6.2 The Most Resource Usage Case of Trainings

7
Conclusion

Image colorization processes are very important to establish a connection between the past and the present. Many methods have been proposed for this task for many years. One of the most recent of these methods is the use of a Generative Adversarial Networks structure. Within the scope of the project, image and video colorization processes were performed with GAN models.

During the project, firstly, the methods used were examined by reviewing the literature and preliminary studies were performed to understand the working principles of GAN architectures. Then, two different discriminators and two different generator structures were created to find the GAN model that gives the optimum performance. Four different GANs were created with these models. By changing the hyperparameters of the models, the performance of 8 different GAN models was measured with the CIFAR-10 dataset and grid search technique. According to the results, the model to be used was decided and based on these models. In the next step, 128x128 pixel 'Hababam Class' and 'Documentary' datasets were created for video colorization. Dependent GAN and independent GAN models were developed based on scene changes in the videos. Finally, an algorithm is developed to detect scene changes and a design is developed for the selection of GAN models. Model performances were measured by SSIM and PSNR scores.

The results show that the colorization process with the GAN structure provides successful results. The potential of the adversarial structure of GAN models in reaching accurate results in the image colorization task is examined. In addition, a new design was developed for video colorization using two different GAN models according to scene change detection.

In conclusion, this study examined the use of GAN models for image colorization tasks and demonstrated their potential for success. It is expected that with the development of models in the future, more detailed and accurate solutions can be developed.

A
Appendings

- Hababam Test Video Result
- Island Test Video Result
- Village Test Video Result

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[2] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, Springer, 2016, pp. 649–666.

[3] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, Springer, 2016, pp. 577–593.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] R. Zhang, P. Isola, and A. A. Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1058–1067.

[6] S. Treneska, E. Zdravevski, I. M. Pires, P. Lameski, and S. Gievska, "Gan-based image colorization for self-supervised visual feature learning," *Sensors*, vol. 22, no. 4, p. 1599, 2022.

[7] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[8] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

[9] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 451–461.

[10] Y. Zhao *et al.*, "Vcgan: Video colorization with hybrid generative adversarial network," *IEEE Transactions on Multimedia*, vol. 25, pp. 3017–3032, 2023. DOI: 10.1109/TMM.2022.3154600.

[11] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[12] L. Kiani, M. Saeed, and H. Nezamabadi-pour, “Image colorization using generative adversarial networks and transfer learning,” in *2020 International Conference on Machine Vision and Image Processing (MVIP)*, IEEE, 2020, pp. 1–6.

[13] B. Lv, Y. Liu, S. Zhang, H. Zeng, and G. Zhu, “Super resolution with generative adversarial networks,” *Association for the Advancement of Artificial*, 2018.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Mehmet Alperen ÖLÇER

Birthdate and Place of Birth: 19.07.2001, Balikesir
E-mail: alperen.olcer@std.yildiz.edu.tr
Phone: 0553 841 79 50
Practical Training: Java and Devops Engineer Intern, 1 Month / Turkcell - AI Research Intern, 2 Months / Institut Jožef Stefan

SECOND MEMBER

Name-Surname: Şevval BULBURU

Birthdate and Place of Birth: 02.03.2000, Istanbul
E-mail: sevval.bulburu@std.yildiz.edu.tr
Phone: 0541 840 39 60
Practical Training: Game Engine Developer Intern, 1 Month / Nitra Game Software - AI Research Intern, 2 Months / Technical University of Kosice

Project System Informations

System and Software: Ubuntu, Python3, Tensorflow-2.16 Library, Tensorflow-io-0.37.0 Library, Scikit-image Library
Required RAM: 32GB
Required Disk: 8GB