REMOVE UNWANTED OBJECT FROM IMAGE/VIDEO

HACER DOĞAN

AUGUST 2021

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER'S THESIS

REMOVE UNWANTED OBJECT FROM IMAGE/VIDEO

HACER DOĞAN

AUGUST 2021

ÖZET

İSTENMEYEN NESNELERİN RESİM/VİDEO ÜZERİNDEN KALDIRILMASI

Hacer Doğan
Yüksek Lisans
Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Dr. Öğr. Üyesi Roya CHOUPANI
Ağustos 2021, 11446

Her alanda hızla gelişen ve ilerleyen teknoloji, görüntü / video işlemede de kendini göstermektedir. Günümüzde bilgisayarla görme, insan görüşüne sahip bilgisayarlara uygulanabilecek önemli bir çalışma alanıdır. Bu alanın konularından biri olan görüntü / video işleme kapsamında nesnelerin sınıflandırılması, tanımlanması, takibi ve bölümlenmesi gibi çalışmalar yapılmaktadır. Video ve görüntü işleme çalışmamız ile istenmeyen nesneleri görüntü/videodan kaldırmayı hedefledik. Bu çalışmada, yapay sinir ağlarını kullanarak tespit edilen nesnenin maskelenmesiyle oluşturulan alanın yani nesnenin kapladığı alana uygun arka plan oluşturmak için üretken düşman ağlar modeli ile uygun piksel setlerini üreterek tamamlamaya çalıştık.

Anahtar Kelimeler: Bilgisayarla Görme, Nesne Algılama, Nesneyi Kaldırma, Üretken Düşman Ağlar, Piksel Üretme

ABSTRACT

REMOVE UNWANTED OBJECT FROM IMAGE/VIDEO

Hacer Doğan

Master of Science

Computer Engineering Department

Supervisor: Assist. Prof. Dr. Roya CHOUPANI

August 2021, 11446

The rapidly developing and advancing technology in every field also shows itself in image / video processing. Today, computer vision is an important field of study that can be applied to computers with human vision. Within the scope of image / video processing, which is one of the subjects of this field, studies such as classification, identification, tracking and segmentation of objects are carried out. We aimed to remove unwanted objects from the image/video with our video and image processing work. In this study, we tried to complete the area created by masking the detected object using artificial neural networks, by producing appropriate pixel sets with the generative adversarial networks model to create a suitable background for the area covered by the object.

Keywords: Computer Vision, Object Detection, Remove Object, Generative Adversarial Network, Generate Pixels

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assist. Prof. Dr. Roya CHOUPANI for giving me the opportunity to work with her and her guidance, motivations and support throughout my master's and undergraduate education.

I would like to thank all my teachers at Cankaya University Computer Engineering Department, for whom I felt their help and learned a lot from them.

CONTENTS

FIGURES

TABLES

ABBREVIATIONS

CNN	Convolutional Neural Network
PDE	Partial Differential Equations
ANN	Artificial Neural Network
GAN	Generative Adversarial Network
RNN	Recurrent Neural Network
PixelRNN	Pixel Recurrent Neural Networks
PixelCNN	Pixel Convolutional Neural Networks
LSTM	Long-Short Term Memory
VAE	Variational Auto Encoders
Pix2Pix	Pixel to Pixel
PROGAN	Progressive Growing of Generative Adversarial Network
DCGAN	Deep Convolutional Neural Network
INFOGAN	Information-theoretic extension to the Generative Adversarial Network
SINGAN	Single Natural Image Generative Adversarial Network
SIMGAN	Semantic Image Manipulation
SSD	Single Shot Detector
YOLO	You Only Look Once
YOLAC++	You Only Look at Coefficients
R-FCN	Region-based Fully Convolutional Networks
Mask R-CNN	Mask Recurrent Neural Network
Fast R-CNN	Fast Recurrent Convolutional Neural Network
Faster R-CNN	Faster Recurrent Convolutional Neural Network

GP-GAN	Gaussian-Poisson Generative Adversarial Network
RESNET	Residual Network
PASCAL VOC	The Pascal Visual Object Classes
BCE	Binary Cross Entropy
ReLU	Rectified Linear Unit
tanh	Tangent Hyperbolic Function
RGB	Red Green Blue
H	Height
W	Weight
N	Number of images
CTS	Crops Training Set
ROI	Region of Interested
STL	Self-Taught Learning
PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Square Error
dB	decibel

CHAPTER 1
INTRODUCTION

Although scientific study fields are divided into different new fields under sub-headings, they are applied as studies that complement each other. This also applies to image processing and computer vision. Then artificial intelligence and machine learning begun to replace the methods developed with some algorithms in solving problems in these areas. On the other hand, deep learning models have started to be very effective in image processing studies. Image and video processing problems are mostly grouped under the following headings. The fact that some of these problems are solved first provides an easy solution for the solution of some other problems. Although the main purpose of this study is to remove objects on the image/frame, evaluation and results on the following subjects are also considered:

- Image processing methods,
- Object detection,
- Segmentation,
- Image repairing,
- Inpainting,
- Completion of the image,
- Pixel creation or estimation,
- Generation of new data,
- Solutions for data needs in limited conditions,
- Methods for obtaining new sub-image.

Among the problems mentioned above, the most accurate determination of the position of the object significantly determines the quality for the removing action in our study. One of the other important problems is what can be done if the data is limited, and the other is the problem of effective data acquisition method.

1.1 SCOPE OF THE THESIS

What the preliminary studies required for object removal on the image/frame and what are these problems among the literature studies will be presented comprehensively.

1.2 CONTRIBUTION

This thesis will briefly describe the methods used in object removal, which has become an important issue in image/video processing, and the evolution of some approaches. In this thesis, the generative and discriminative models will be implemented to replace the object/s, which find an important area of use today. This study will also demonstrate an important method of obtaining necessary and useful data to build an effective generative and discriminative model.

1.3 STRUCTURE OF THE STUDY

Remove an object is a practice from a video/image one of subject of image/video processing. Removing an object is an application from video/image which is one of the image/video manipulation topics. This includes a process, from detection, recognition of the object to the insertion of data to replace the object. This process will be evaluated under various headings from the beginning until a new visual data is generated and saved. Overall structure for this thesis is as follows:

In Chapter 2, subjects such as the recognition of the parameters to be used in the detection of the object, segmentation methods, classification concepts, producer networks and definitions, types according to usage will be explained. Finally, the subject of filling the empty spaces that occur when the detected object is removed will be mentioned.

In Chapter 3, what are the studies done in object detection, object detection methods with traditional methods using various algorithms, and a literature study of segmentation, classification and detection studies based on deep learning models, which are more preferred recently. Likewise, while trying to find pixels to replace the object, the methods used from the past to the present will be mentioned. A literature review will be presented to obtain new pixel clusters with generative models, especially in recent times.

In Chapter 4, object detection and segmentation models used in this study will be discussed. The structure of the generator model used, generator and discriminator networks will be explained in detail.

In Chapter 5, information about the characteristics of the data used in the study and data sets will be given. The approach of obtaining the data set will be mentioned and our experience and results will be mentioned. The libraries used will be briefly mentioned and the implementation methods will be shown. In this thesis, some of the results we reached will be presented in comparison with some other articles.

Chapter 6 includes a summary for our work in this thesis and finally, it was thought that the suggestions for increasing the study efficiency of this study and improving the produced result could be tried in the future.

CHAPTER 2
BACKGROUND

In this section, knowledge about which problems in the background of this study or what needs to be done first to complete the study will be explained. Firstly, since it is necessary to detection and identification of the object, the terminology and details of this subject will be explained. Then, what we need to know about generative models will be summarized. Finally, it will be explained what the approaches are related to image inpainting and completing the missing part on the image/frame.

2.1 OBJECT DETECTION

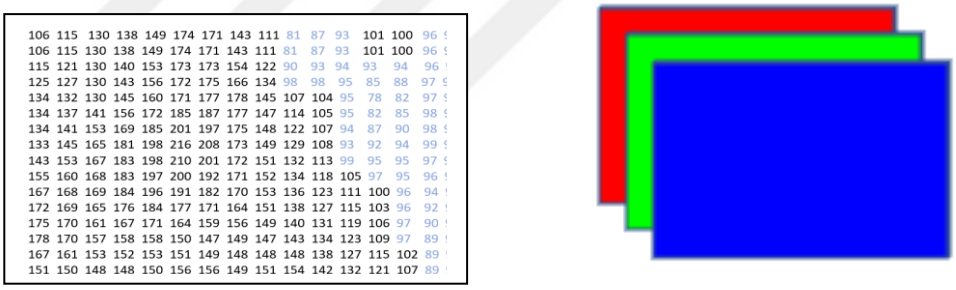


Figure 2.1: On the left, partial object on image (just blue layer) with computer vision, on the right, an image illustration with 3 layers (blue, green, red).

The object definition defines each element class on the image/frame, such as a human, a bicycle, a cat, a car, in this study. A video is a sequence of images that is shown in succession at a given time. For this since processing is done on each frame video processing can also be said as a series of image processing for this study.

Video processing subjects such as object recognition, detection, location determination, segmentation and classification have an important place in image processing and computer vision studies. Segmentation methods differ. The importance is increasing according to the area of use, for example, locating objects in satellite

images in areas such as in geography as macro level [1] and detecting structures in medical images for health as micro cell level [2], segmentation can be more detailed and may be of massively importance in the most accurate location and definition of the object. Making the most accurate distinction between similar structures increases the percentage of accuracy in solving problems and getting the meaningful results. Image segmentation techniques in the literature are threshold method, region-based segmentation, edge detection segmentation, cluster-based segmentation, convolutional neural network (CNN) based segmentation, watershed-based, and partial differential-based method [3] [5] [6] these will be briefly mentioned below.

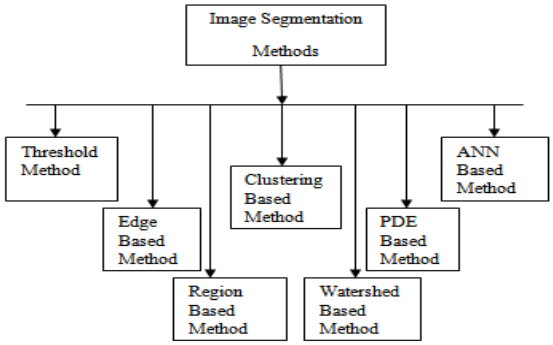


Figure 2.2: Segmentation methods [5].

Threshold method based on the histogram peaks of the image to find required threshold values [5]. Pixels divide by compare intensity with threshold value

Region-based segmentation is similar as threshold segmentation. This approach is based on detecting similar pixels in an image that based the region growing, spreading, and merging with the specified threshold.

Edge-detection segmentation is based on discontinuity detection unlike pixels detection. If image have better contrast between objects and backgrounds this approach will be effective but not suitable for noisy images [4].

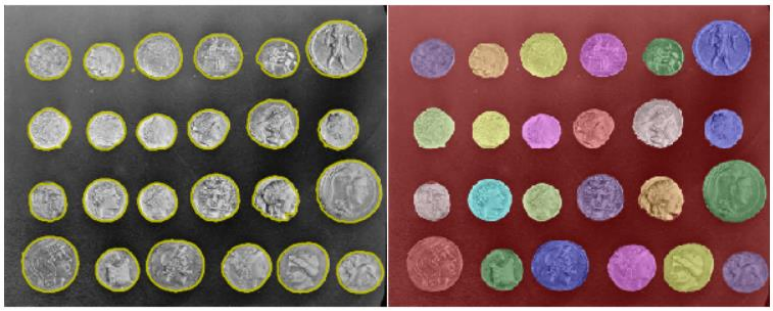


Figure2.3: Edge detection vs region segmentation (from scikit-image)

Cluster-based segmentation based on division into homogeneous clusters [5]. Clustering algorithms are unsupervised learning. The most well-known clustering algorithm was developed by J.B. MacQueen, is the k-means algorithm. Fuzzy clustering (also referred to as soft clustering or soft k-means) is a form of clustering. Fuzzy c-means (FCM), also known soft clustering or soft k-means, clustering was developed by J.C. Dunn in 1973 [8] and improved by J.C. Bezdek in 1981 [9].

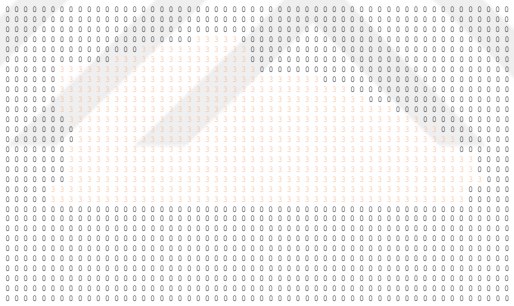


Figure 2.4: Partial object mask.

Watershed-based method is morphological image processing technology that based on topological interpretation, detected boundaries are continuous [7]. Also, it is an approach of region-based method.

Partial differential-based method based on simulation of learning process for decision making, useful for best critical applications. The advantage of this approach consists in the possibility of segmenting highly noised images. [5,6]

CNN based segmentation based on simulation of decision making [5,10]. There are many different approaches for the decision object recognition, identification to segmentate the image/frame [11,12,13,14,15,16,29]. Although, this approach more

wastage of time in training, high accuracy results and it is preferred because there is no need for complex programs. The objects and classes learned from previously trained datasets can be used to estimate by transfer learning. Hence, time consuming eliminated. Computer vision also deals with the object from different perspectives, working on image segmentation. This can define as image segmentation type that include instance segmentation [17,18,19,26,27], pose estimation [20,21,22], panoptic segmentation [23], key-points [24], semantic segmentation [25,28,30,82]. Other work topics related to the object are tracking the object, background subtraction [55], foreground extraction [54], changing the property of the object, or removing the object. In this study, deep CNN-based method is used to detect instance segmentation for removing the object from image/frame.

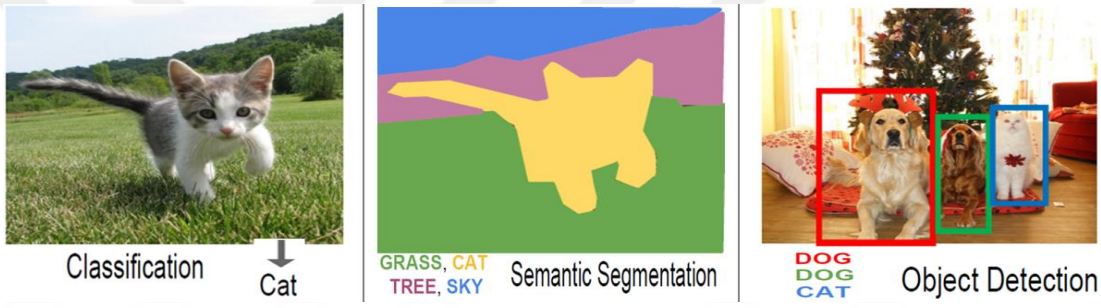


Figure 2.5: Classification, segmentation, and object detection.

2.2 GENERATIVE MODELS

A generative model learns from data distribution, density estimation in unsupervised learning. It means that they can produce new content. Generative models learn true data distribution of training set to generate a new data variation. Generative models work via principle of maximum likelihood, but not every generative models.

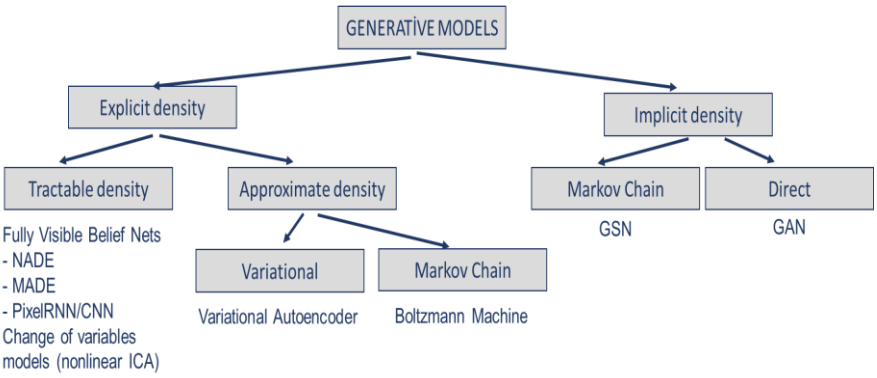


Figure 2.6: Generative Models copyright and adapted from [32].

2.2.1. PixelRNN and PixelCNN

PixelRNN and PixelCNN defines tractable density function, for optimizing likelihood of training data. It defines explicit density function (1) Pmodel (x; Θ). (Θ: parameters).

$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, \ldots, x_{i-1}) \quad (1)$$

Likelihood of image x

Probability of i'th pixel value given all previous pixels

The model assigns to the training data, complex distribution over pixel values. PixelRNN [33] generate image pixels starting from corner, dependency on previous pixels that uses RNN(LSTM) model. PixelCNN [34] generate pixels, too. Start to generate from corner but dependency on previous pixels but uses a CNN model over region of context. Both models use same function to maximize likelihood of training images.

2.2.2. Variational Autoencoders (VAE)

VAEs unsupervised and deep generative model of Autoencoders (AE). AEs cannot to generate new data and thanks to VAEs can make regularization, versions of autoencoders making the generative process possible. Standard Autoencoders and VAEs have encoding and decoding but Standard Autoencoders can regularization after decoding data. VAEs can make input data reconstruction. Encoder and decoder networks also called *recognition/inference* and *generation* networks.

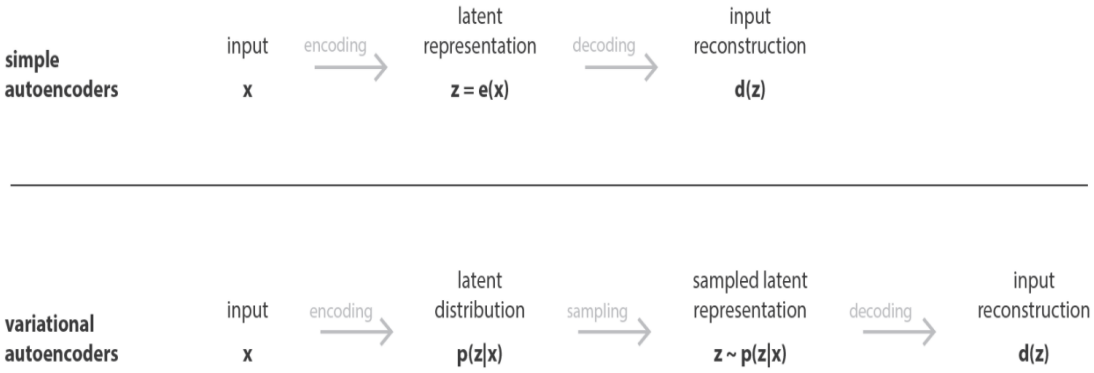


Figure 2.7: Difference between autoencoder (deterministic) and variational autoencoder (probabilistic). (Image taken from [93].)

VAEs define intractable density function (2) with latent z:

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz \qquad (2)$$

This cannot optimize directly, instead of it, derive, and optimize lower bound on likelihood. VAEs learn the complicated data distribution like images in unsupervised learning. Model learns probability distribution of training data. Training data (latent variables) can store useful data that model try to generate. VAEs learn both the generative model and an inference model.

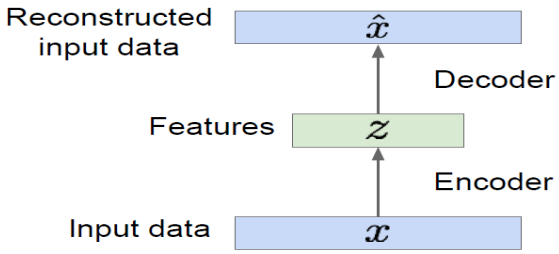


Figure 2.8: Autoencoders can reconstruct data and can learn features to initialize a supervised model.

VAEs differ from GAN in the training way. Now let's look GANs.

2.2.3. Generative Adversarial Networks

Generative Adversarial Networks (GANs) are announced by [31] as a new type of deep generative networks. GANs are used various problems generating human faces [35,42], image-to-image translation(pix2pix) [36], style transfer [37,41], semantic manipulations [38,39,40], photo blending [41,43], resolution solutions [43,44]. GAN belongs to set of generative models. GAN model can use any set of data, but data represents image/frame as a set of pixels, in this study. There is a training data set with real data in GAN models, as in every deep neural network. A GAN model consists of two different network model. A discriminator and a generator. Networks start to work with a generated complex random variables from simple uniform random variables that is a N dimensional vector. Discriminator model decides with probability distribution whether it is a real or fake result according to training set. This result is new input for generator model, the generator model tries to minimize loss according to updated results. The cycle begins between generator and discriminator until the very close to real data that result is produced.

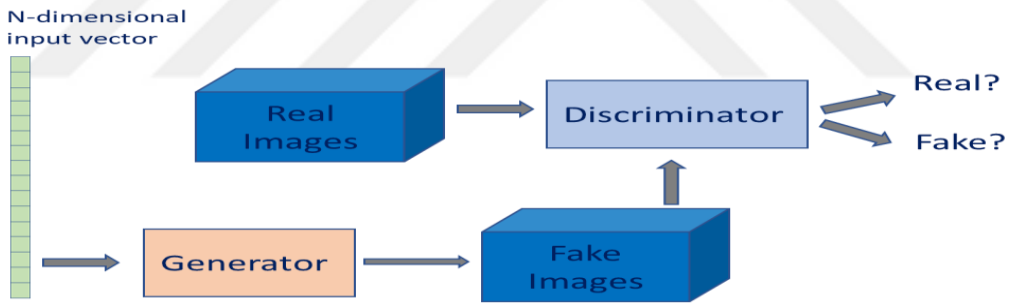


Figure 2.9: Simple generative adversarial network structure

- Generator

The generator models a transform function. Generator takes as input a simple random variable and return trained result, as targeted distribution. The aim of the generator is to fool the discriminator, generator network is trained to maximize the final classification error between real and produced data.

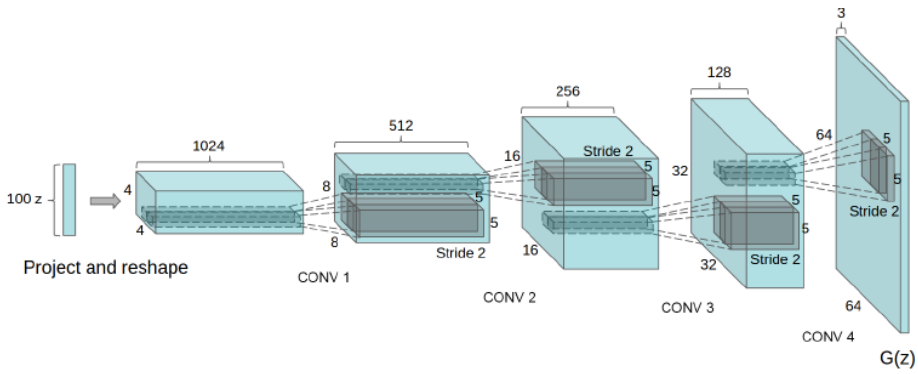


Figure 2.10: DCGAN generator structure. (Image taken from [46].)

- Discriminator

Discriminator models a discriminative function, takes as input (N dimensional vector) and returns as output the probability of true data. Discriminator network is trained to minimize the final classification error real or fake class.

Generator function is represented G(z). ("z" is random generated data). Discriminator is represented D(G(z)). Both are iterated together as in the following equation (3):

$$\max_{G}\left(\min_{D}E(G,D)\right) \quad (3)$$

These two networks together named as Generative Adversarial Network. Classification error is metric for training of generator and discriminator network. There are some types according to the structure of the network such as VanillaGAN, DCGAN [46], InfoGAN [45], WassersteinGAN [47], ProGAN [35] SimGAN [57]. In this study, a GAN model [46] used to generate a sub-image (like a patch) to displace which removed object from content.

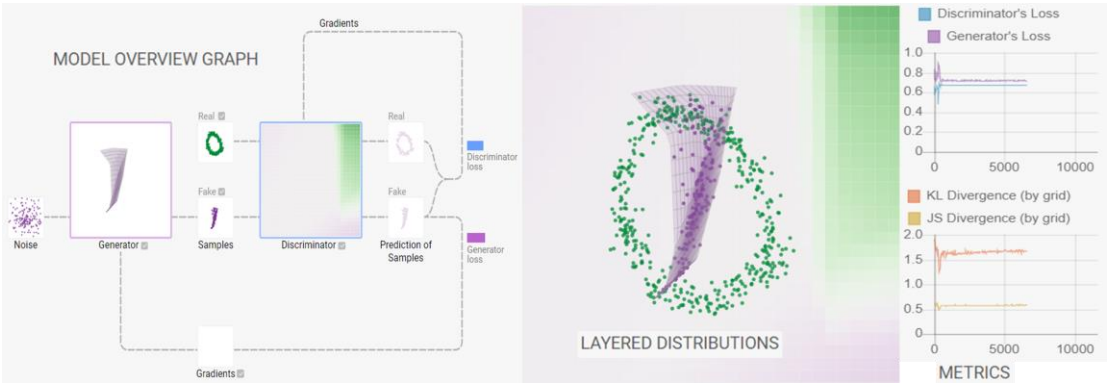


Figure 2.11: Model graph and data distribution with simultaneous loss functions generator and discriminator, iteration around 7500. (Results taken from GANLAB [93].)

2.3 VIDEO INPAINTING AND IMAGE COMPLETION

The main purpose of video and image/frame completion, repair, inpainting is to complete or to fill in missing regions, the gaps on the contents so the areas that need to be repaired, without breaking the integrity. The work of undesired object removal will be studied in this context, in this thesis.

CHAPTER 3
RELATED WORKS

3.1 OBJECT DETECTION

Nowadays, data acquisition and storing is increasing rapidly, and the concept of big data and deep learning has entered our lives with it. This is true for image data, as in every field. There are tens and hundreds of cameras in our living spaces. Images taken from cameras for various purposes, especially for security purposes, are obtained on maps, navigation systems, roads, and streets. The presence of images of individuals in images used for such purposes may be unnecessary and even a violation of personal privacy and security. It is also undesirable for our license plate to be visible when we visit natural and historical places. Or sometimes, we may want people and some unwanted objects to be invisible when we are just making a visual presentation of a place. Computer vision and image processing studies have also started to deal with the removal of this and similar unwanted objects from the image. The removal of the unwanted object is primarily possible by determining the position of the object on the image in the most accurate way. Traditional object removal methods have now begun to give way to deep learning methods since the increase of data and successful results of machine learning methods began to be observed. Ever since it was decided to use big data and for what purposes, companies and computer labs train some data in advance and make the trained models available to people. Thus, the concept of transfer learning is in usage. By using pre-trained and computer-known predefined object classes, object detection [17,82,86] is easily achieved. This has almost eliminated the time-wasting problem. In this study, it was analyzed which studies were conducted in the literature for the fastest and most effective use of object detection and masking. Some of these studies for object detection and identification methods used in this study were examined in detail. Because minimizing the loss of time is also an important goal.

Various methods have been studied to detect objects and determine the object class which means identification. Some of these studies consist of studies that were previously done using algorithms. With the increasing data and deep neural learning methods, these studies have evolved into machine learning methods. Some of these studies can be summarized as follows.

Edge detection techniques for segmentation [3] is the most familiar approach to detect significant discontinuities in density values. In this approach, areas and regions are determined by detecting the edges, and the object class cannot be determined. It is known as region-based segmentation approach [4]. Edges occur on the boundary between two regions.

The most known edge detection methods for image segmentation in the literature Roberts edge detection, Sobel Edge Detection, Laplacian edge detection, Kirsh edge detection, Robinson edge detection, and Canny Edge Detection. Edge detection algorithms are mostly used to find boundaries by filtering the integer values of vertical and horizontal iterating over whole pixels.

Another object identification method is fusion based object detection [49] and this approach that is considered a clustering method. Accordingly, the individual local parts of an image that have the important properties of the object contain the necessary properties for the whole object. A genetic algorithm is used to combine all local and global features, and the algorithm is tested on a set of real-world images to detect objects.

In this work that is named as "Color Image Segmentation Based on Watershed Transform and Feature Clustering [7]", they proposed a novel image segmentation algorithm by combining the watershed transform and feature clustering. This work is a morphological image processing technology. The input image is pre-processed and find details. Then, a marker-based watershed transform applies to segment the image into watershed regions, with mean shift algorithm exploits to cluster the watershed regions segmented.

Another study [51] provides an approach for efficient object detection and matching in images and videos with feature classification. Features extracted by a classification scheme are classified into object features and non-object features. This

classification scheme is a binary classification for object detection and matching. It is thought that this approach can be used for real-time object tracking and detection.

A real-time object tracking study [48] is a color-based probability matching study for real-time multi-object detection. With this study, it is aimed to detect the moving object(s) and track the same object(s) appearing in the next frame. First, the object is tried to be detected with the background subtraction and optical flow method, and the frames are compared to detect the moving object(s) and follow the same object(s) appearing in the next frame.

Image Segmentation using partial differential equations (PDEs) is a popular technique in image processing. With a concise comparison [6], Sliz and J. Mikulka present a survey of state-of-the-art segmentation methods that exploit partial differential equations, focusing on techniques introduced since the year 2010. Image segmentation via PDEs introduce the partition of the image into areas corresponding to objects located in the image. Certain object parameters, such as the type of boundaries, color range, or textures, can be described by a PDE. Active contours and level set methods are used exploiting the parametric curve evolution described by PDEs.

Selective search [50] approach is region-based finding locations and define a bounding box around object. It combines exhaustive search and segmentation methods. This method separates different objects in the image by coloring them different colors.

Region-based convolutional network methods such as Fast R-CNN [12] and Faster R-CNN [13] for object detection. These builds on previous work [87,88] to classify object proposals using deep convolutional networks. Faster R-CNN is faster than from Fast R-CNN with a simple alternating optimization.

Known as R-FCN [14] is a region-based fully convolutional network. It is a "Region-based detector that is fully convolutional with all computation shared on the entire image. Network model applies a costly per-region subnetwork hundreds of times." The network consists of shared, fully convolutional architectures [82]. All learnable weight layers are convolutional and are computed on the entire image. R-FCN ends with a position-sensitive ROI pooling layer. This layer aggregates the outputs of the last convolutional layer and generates scores for each ROI. The R-FCN

architecture is designed to classify the ROIs into object categories and background. [14].

Mask R-CNN [18] is a framework for object instance segmentation. With this approach objects are detected and generated a segmentation mask for each instance. This method extends Faster R-CNN, for predicting an object mask in parallel way for bounding box recognition with the existing branch.

SSD is a method for object detection with a single network. Known as "Single Shot Multibox Detector for multiple objects. Tries to find true bounding box around each object. When prediction time network to generate scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape and network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes "[15].

You Only Look Once (YOLO) is a single neural network and a real-time object detection to predict bounding boxes and class probabilities. YOLO [27] model "Can process images in real-time at 45 frames per second". With YOLO model, input image resizes to 448 × 448, runs a single convolutional network on the image, and thresholds the resulting detections by the model's confidence. Model learns general representations of objects.

Bolya et al. [26] define their model YOLACT++ as better real-time instance segmentation. It is a fully convolutional model for real-time (> 30 fps) instance segmentation. Model works on two parallel subtasks first generating a set of prototype masks then predicting per-instance mask coefficients while instance segmentation process. This approach produces high quality masks over object detected on image/frame.

Recent work in object detection has mostly been on the development and refinement of previous work. One of these studies is the object detection method called Lightweight -YOLOv3 [95]. With this study, it is aimed to reduce the model size and computational complexity by developing the YOLOv3 [27] network architecture. According to this architecture, vehicle and object detection has been studied.

3.2 REMOVING UNWANTED OBJECT(S) IN IMAGE/VIDEO

The image consists of a number(s) objects and content that can be defined as the background. The subject of determining the object and background from the image, removing object has been the subject of image processing in some previous applications [76].

Video inpainting goal is removing objects or repair missing, defective regions in a video sequence by using locational information of the object from each neighbor frames with preservation of visual consistency. Sometimes we may want some objects not to be present in the visual recordings that obtained for various purposes. We may not want to include certain objects (like someone else) in our daily visual data or may want to respect personal privacy for others. An approach can be made as incomplete content completion in old image data. As the main purpose, there are study subjects such as image inpainting, image completion, restoration, generating pixels, predicting pixels to fill the gaps and deficiencies in visual data due to various reasons. It is aimed to produce appropriate pixel sets to replace the removed object with the generative method. Gao et al. method [77] is named flow-edge guided video completion. Their method firstly extracts and completes motion edges, and then use them to guide piecewise-smooth flow completion with sharp edges and non-local flow connections to temporally distant frames, enabling propagating video content over motion boundaries [77].

Various algorithms were used such as [52,53] before generative approaches were widely used with methods such as artificial intelligence and deep learning. The literature review for this is as follows.

The concept of inpainting is more commonly known as a term used in the restoration of painted images and it was historically done manually by painters for removing defect from paintings and photographs. Image inpainting algorithms have different approaches such texture synthesis and exemplar and search base inpainting as the first approaches [89] in the literature. Other inpainting which uses FMM (Fast Marching Method (Telea algorithm)) [63], another used inpainting method is with 2D NSE (Navier-Stokes Equations) [90] for image. These methods are not suitable for filling large gaps in the image. The other one is described as shift-map editing by Pritch et al. [91].

One of the algorithms for removing objects from video is an approach that works as tensor completion. Accordingly, there is no need for any information about the object, instead it is sufficient to remove all information about the unwanted object by replacing the unwanted pixel value with zero. The method assumes, the tensor to be recovered forms a low-rank structure [56].

Patch-based approaches have also been effective in some studies. Huang et al. [66] have used constructs such as perspective and regularity to guide the filling of missing regions in a semantically meaningful way. We can say that this is a patch transformation model. This work can complement scenes such as multiple facades, large regularly repeating structures. The patch should be an image that is very similar to the nearest neighbor of the area to be filled. Appropriate patch should be selected for the correct reconstruction of textures and structures. Newson et al. [70] preferred minimization of a patch-based non-local functional. They first search approximate nearest neighbor, then with a reconstruction processing and inpainting the image with textures.

The term mask is often used in the process of removing an object from an image or video. Defines the repainting or coverage of pixels containing the location of this object. The most important part of the removal process of an object often starts with the mask definition. The other part is completed with the pixel values to be recalculated, predicted, or find the correct local fill of the object area. Generative models use a mask to detect the unwanted object location, mostly. Filling concept with neural networks and deep generative models for inpainting and removing the object literature as follows.

Generative methods and deep learning-based models for image completion or inpainting:

The human brain can describe a whole object with its partial image. Based on his previous experience, we can predict this with the information about the structure of the object in his memory or the missing part of the image from the whole object. Similar this idea, Pathak et al. [59] extracted features from the surrounding context, used a convolutional neural network which called context encoder model to find missing pixel values when given an image with a missing region. The context encoder

has similar encoder-decoder architecture. It is a kind of parametric inpainting algorithm that can fill missing region with logical semantic results [59].

Described as CC-GAN model which context- conditional generative network has similar approach [59] that is a semi-supervised learning model for images based on in-painting using an adversarial network. A generator fills the hole, based on the surrounding pixels on images with random patches removed [73]. Their model can evaluate on STL-10 and PASCAL datasets.

Semantic inpainting generally use for image that includes one object (e.g. a face, a car, a number) with a partial absence. Yen et al. [72] using data of this feature arbitrary masks without re-training the network on the object, unlike of using a central mask [59], it generates the missing content by conditioning on the available data, the missing content by conditioning on the existing data with the DCGAN model. Another semantic inpainting method similar [59,72] but there are some
differences with "A multi-scale neural patch synthesis approach based on joint optimization of image content and texture constraints, which not only preserves contextual structures but also produces high-frequency details by matching and adapting patches with the most similar mid-layer feature correlations of a deep classification network" [65].

In some cases, images need to be completed from the full scene and not just part of the objects. Some studies are also aimed at this purpose [62,67,61]. An incomplete and irregular area may be found, or it may cover an arbitrary location on a picture, to complete the integrity of the picture and obtain the realistic picture, as in these works. Mask for inpainting regions can be free-form, the work [61] help user to remove unwanted objects. It is a study [68] that aims to fill the gaps in the old manuscripts with worn and irregular deficiencies to complete the semantic integrity with a similar purpose in a different field.

Although progress has been made in image processing, studies have mostly been done on object-centered images such as faces or structured scene datasets. On the contrary, this study is a study for the removal of an object directly. Shetty et.al [58] describe scene editing and their work is an automatic interaction-free object removal model. Since this study is like our study purpose, comparisons were made in the following ways Figure (5.5,5.6).

Li et al. [69] worked on a deep inpainting with fully convolutional network model that analyze the difference between inpainting region and other regions, using high pass filtered image residuals.

The multi-scale network approach with the high-resolution inpainting technique [64], which is a method for the problem of the image in daily life which include unnecessary pedestrians in image background. This network removes unwanted pedestrians from the image, which is detected instance result by Mask R-CNN [18].

In a study carried out to complete the missing parts of the object and restore its integrity, first, the method of determining and completing the edges of the object via edge connect generator and then the generative inpainting method was applied [60]. Like inpainting with an edge connect generator, structure reconstruction and texture generation are a two-stage model that completes the missing structures of the input image [75].

Various results can also be produced by blending two different images. This is what we can call high-resolution, well-blended composite images. For this purpose, Wu et al. developed a generator network model (GP-GAN) to obtain new realistic images. This approach can also be used for image-to-image translation [43].

Video completion efforts may be more effective than image completion in terms of acquiring data between frames, due to the need to preserve temporal information and consistency, or with some difficulties related to optical flow. In video completion, the use of data between frames makes it closer to reaching the required data and increases efficiency in target-oriented results. The studies mentioned, so far, were in the form of accessing the required data with object completion oriented external datasets or aimed at background completion with less complexity. It aims to complement with distant interframe data for video completion. One of them is an internal learning approach [78] that is a network while training on a masked input video without any external data. The other one is a copy paste deep neural network [79] that is also used between distant frames data, the network is trained to copy corresponding contents in reference frames and paste them to fill the gap in the target frame.

A work is an object removal and inpainting with two generators that is a combined GAN. One generator removes the object other fills the background. The network removes the desired object from an image at once without any separate object recognition process [71].

The scope of the subject of changing the semantic content in the picture has increased. Like we want to remove an object, we can add an object suitable for an environment or structural changes can be made instead of changing only the color-related content of a structure. It is a work [40] that includes semantic image editing tasks with neural networks in the image content. In visual content, it can focus on some objects and ignore the background or separate the object from the background. Studies for this purpose, known as foreground extraction background subtraction, in image processing are also available in the literature [54,55].

More relevant works of image inpainting or completion is conditional image generation methods that transform images in different domains between paired or unpaired data. Many studies have been done since first generative model was published, continuing to evolve for various purposes based on [31]. This change was originally designed to train and generate a particular class as conditional generative models. Then, models such as generators and transformers are used that can make semantic changes of some features of one image on another image. (C-GAN [94], Pix2pix [36], Pix2pixHD [38], Cycle-GAN [41].)

C-GAN [94] is a model that conditionally generates samples for only one or a few desired classes among large data sets.

Pix2pix approach [36] synthesis photos from label maps reconstruct objects from edge maps and colorize images. It is a generative adversarial image-to-image translation method with conditionally, based on [94]. Another conditional generative approach is to synthesize and to manipulate semantically for high resolution images (pix2pixHD) that uses object instance segmentation information, enables object manipulations such as removing/adding objects and changing the object category and edit the object appearance [38].

And lastly, Cycle-GAN [41], on the other hand, aims to convert a pixel between the desired classes between two different unmatched images that is, “The model learns the match between an input image and an output image using a training set of aligned

pairs of images.” The most well-known examples in the literature in this study are the conversion from horse to zebra, from zebra to horse, from summer to winter, from winter to summer. It is a work that based on pix2pix [36].

In this recent study [96], the images of cars on the road were removed by using the mask data with dilation obtained by the data augmentation method and the Mask R-CNN [18] method.

CHAPTER 4
MODEL

This section describes the approach to removing the object. This process includes determining the coordinates of the pixels representing the object to be removed and producing and placing new pixel sets to replace these pixels. In other words, the area where the object to be removed is defined as missing areas and it is tried to produce realistic complementary alternative pixels that complement the visual. Deep learning models that enable to identify predefined object classes and libraries used for transfer learning in this study will be mentioned. The required pixel production will be explained. The structure of thesis is shown Figure (4.1). In this study, Pytorch [86] libraries used to detect object classes on the content, mostly but manual mask created for some sample to identify the object location Figure (5.2, 5.6(c)).

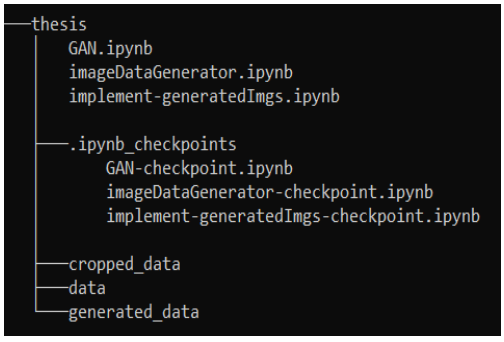


Figure 4.1: File tree of the thesis

4.1 RESNET

Because of difficulties of deep neural networks to train, announced a new residual learning framework [80] to ease the training process. RESNET architecture defines as residual convolutional neural networks for image classification tasks. It was

first winner in the ILSVRC-2015 classification, COCO detection, COCO segmentation, ImageNet detection, and ImageNet localization tasks. Increasing number of layers in deep neural networks cause the training and test error rate also increases. It shows as Figure (4.2) [80].

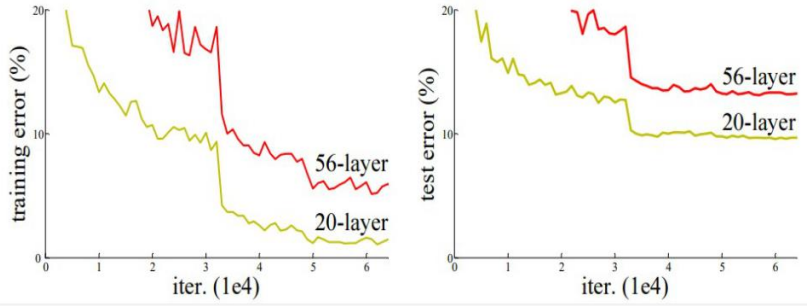
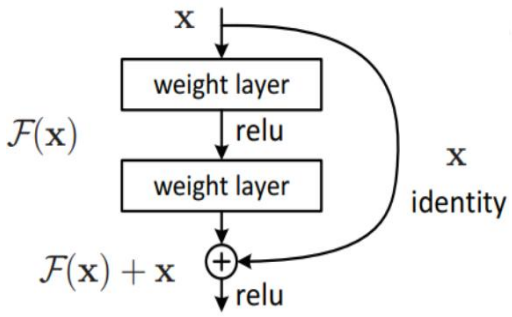


Figure 4.2: The figure taken from the RESNET [80].

These errors related vanishing/exploding gradient problem. To solve this problem used a technique skip connection. Skip connection means connecting directly to the output between x input to the output after few weight layers in training process as below.



The output H(x)= F(x) + x. The weight layer learns a kind of residual mapping: F(x)=H(x)-x. Hence, if there is a gradient problem in a layer, network has identity x to transfer previous layers.

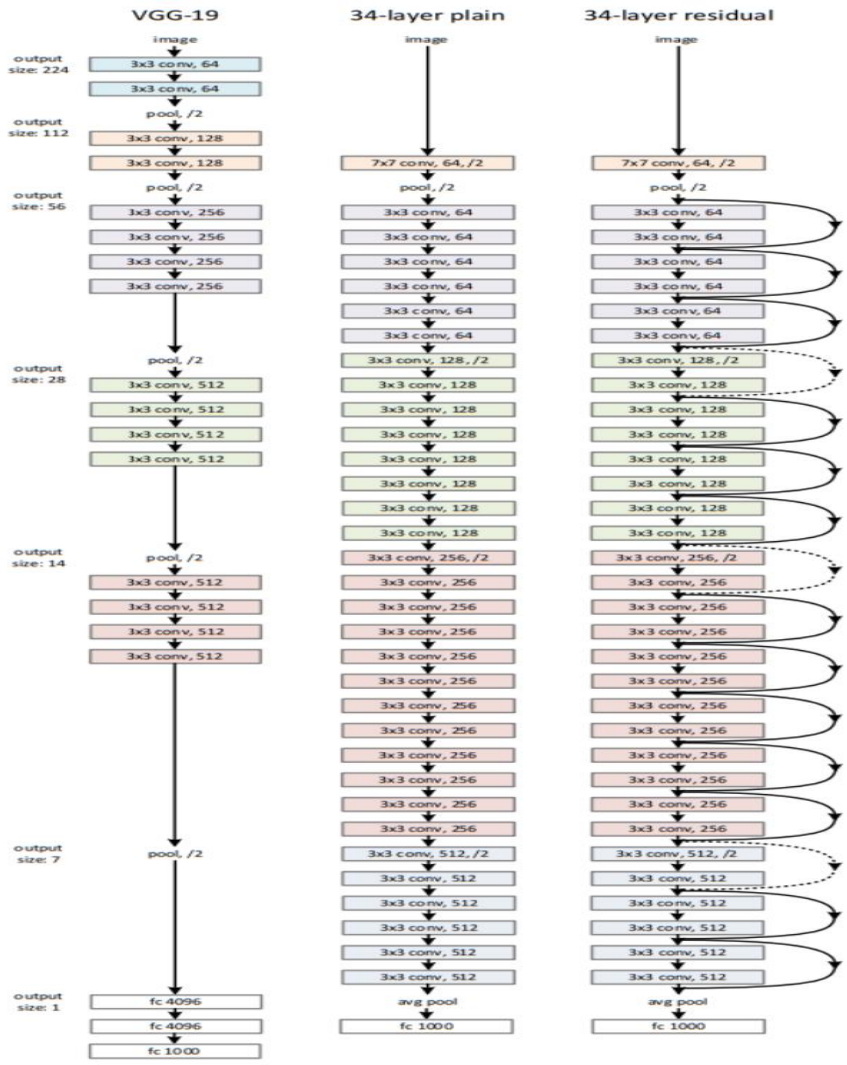


Figure 4.3: Resnet architecture with 34-layer residual [80].

There are types according to the number of layers, ResNet18, ResNet34, Figure (4.3), ResNet50. ResNet101 and ResNet152.

4.2 DEEPLABV3-RESNET101

Deeplabv3-ResNet101 is constructed by a Deeplabv3 [17,30] model with a ResNet-101 backbone. The pre-trained model has been trained on a subset of COCO train2017, on the 20 categories that are present in the Pascal VOC dataset. Global Pixelwise Accuracy of the pre-trained models is %92.4 on COCO val2017 dataset. The model uses pre-trained model to predict and plot semantic segmentation result of 21 classes in each color. Deeplabv3-ResNet101 published by "Pytorch Team" and available from [81].

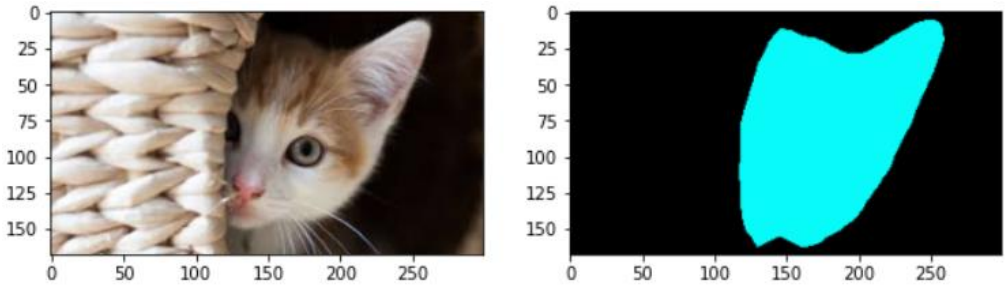


Figure 4.4: Deeplabv3-ResNet101 model defined a mask pre-defined object class with shape (heightXweightX1) image data (Image taken from Google).

4.3 FCN-RESNET101

FCN-ResNet101 is constructed by a Fully Convolutional Network [82] model with a ResNet-101 backbone. The pre-trained models have been trained on a subset of COCO train2017, on the 20 categories that are present in the Pascal VOC dataset. The model use pretrained model to predict and plot semantic segmentation result of 21 classes in each color. FCN-ResNet101 published by "Pytorch Team" and available from [83].

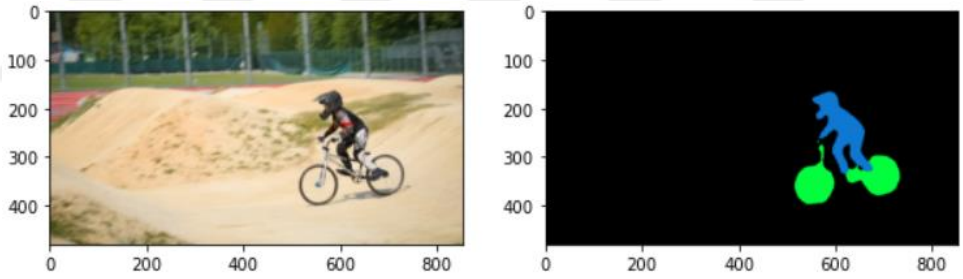


Figure 4.5: FCN-ResNet101 model defined a mask pre-defined object class with shape (heightXweightX1) image data. (Frame from DAVIS dataset bmx-bumps.

4.4 GAN

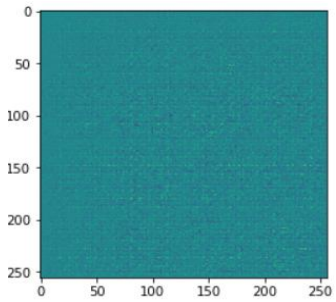


Figure 4.6: A random input generated via generator model (256x256x3) according to(1x100) vector.

GAN model, build with Tensorflow [84] library for DCGAN [46] model that re-build to generate Figure (4.9) (Nx256x256x3, N is number of image set) pixels set. Tensorflow DCGAN generate (Nx28x28x1) samples. GAN model starts with an input vector containing random numbers representing 1x100 Figure (4.6) pixels. The input vector from random generator goes to the discriminator and continues to produce a false result. The discriminator compares by looking at the training set and produces a fake result for the specified number of cycles. On the other hand, the generative network generates a possible new vector by increasing the similarity ratio according to the real data distribution according to the output value returned by the discriminator. This vector is the generated output. The generator model produces the desired number of images in the size of 256x256x3. The GAN model used is as shown in the Figure (4.7), convolutional kernel is 3x3, strides are (2,2), ReLU function used for each layer generator and discriminator. Generator model used batch normalization and tangent hyperbolic function (tanh) as activation function and discriminator used %0.3 dropout. “Binary Cross Entropy (BCE)” function is used for both generator and discriminator loss to compute cross entropy. Binary cross entropy calculates the difference between two probability distributions. Adam [85] is an adaptive learning rate optimization algorithm, designed for training deep neural networks.

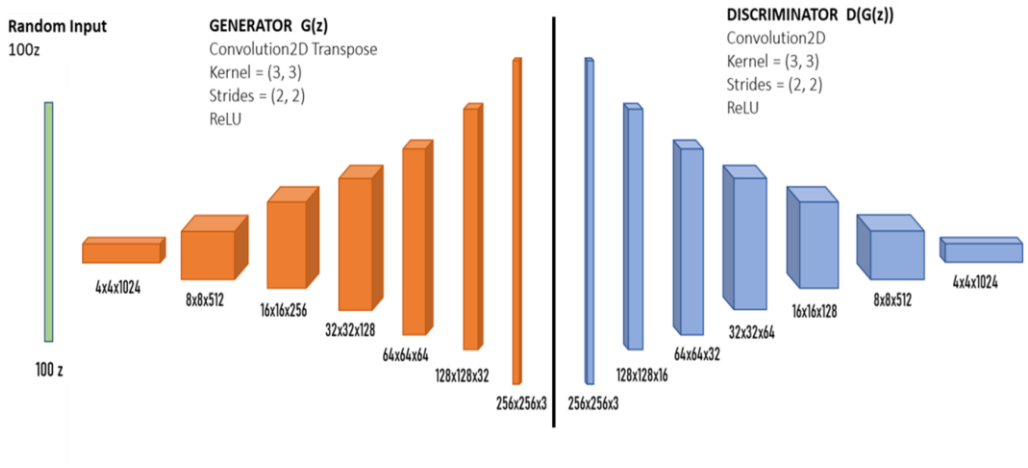


Figure 4.7: GAN model used, the illustration of up-sampling (Convolution2D Transpose) and down-sampling (Convolution2D).

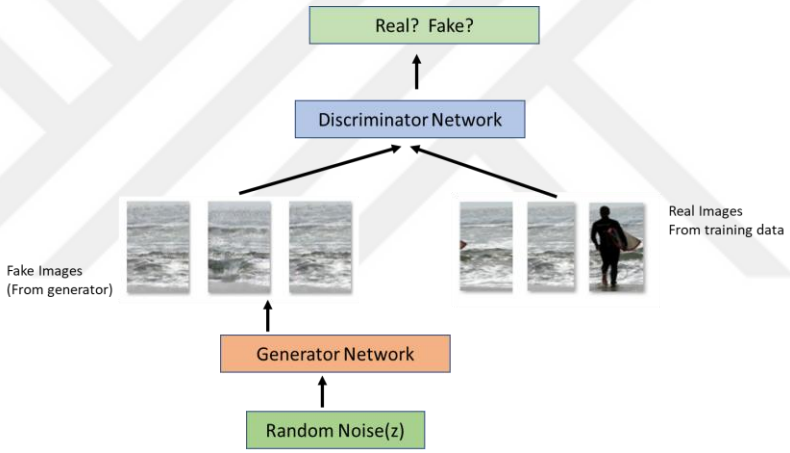
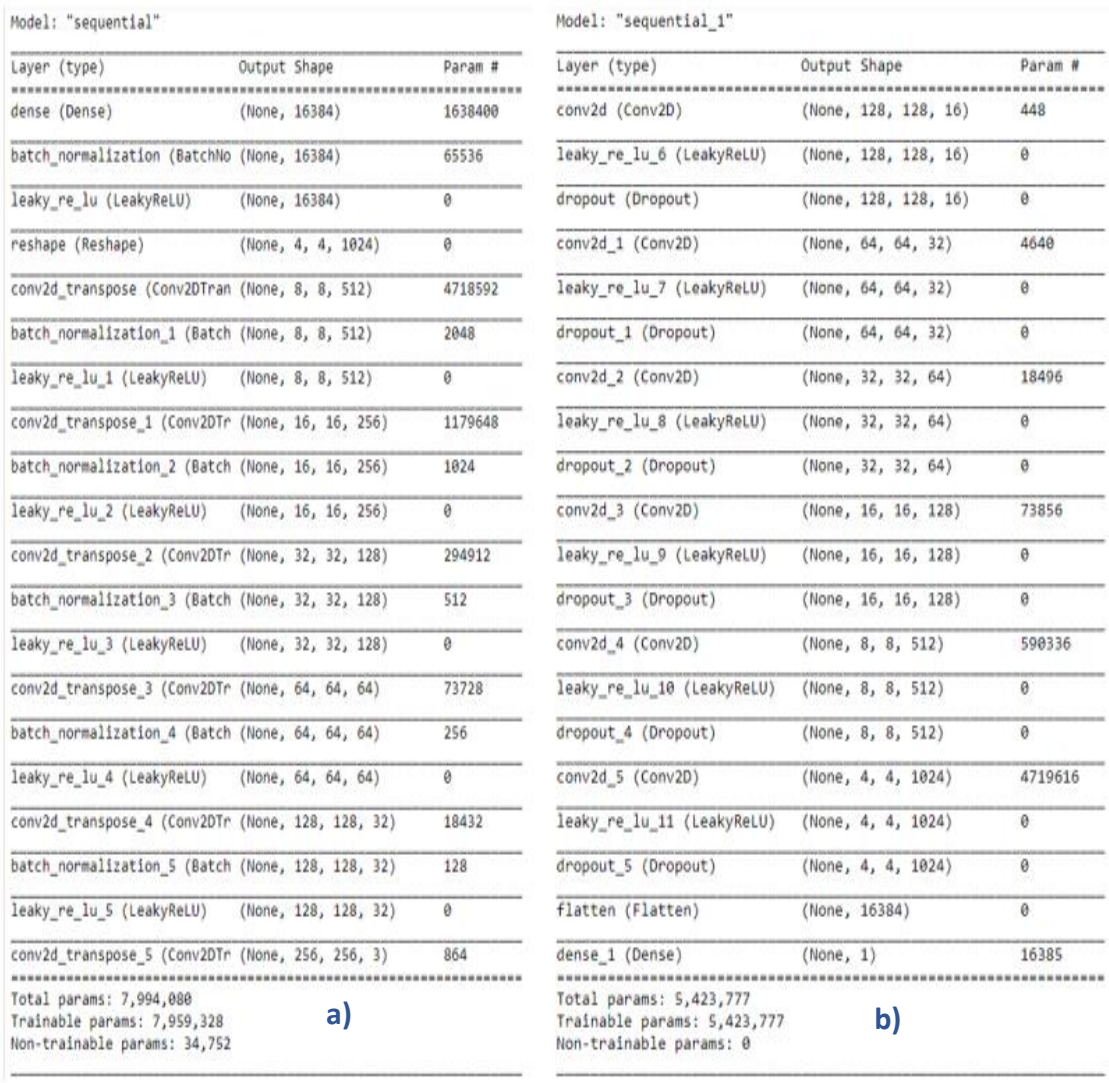


Figure 4.8: Discriminator and generator produce new images that are like the images in the model training set. (Image from COCO dataset)



Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 16384)	1638400
batch_normalization (BatchNo	(None, 16384)	65536
leaky_re_lu (LeakyReLU)	(None, 16384)	0
reshape (Reshape)	(None, 4, 4, 1024)	0
conv2d_transpose (Conv2DTran	(None, 8, 8, 512)	4718592
batch_normalization_1 (Batch	(None, 8, 8, 512)	2048
leaky_re_lu_1 (LeakyReLU)	(None, 8, 8, 512)	0
conv2d_transpose_1 (Conv2DTr	(None, 16, 16, 256)	1179648
batch_normalization_2 (Batch	(None, 16, 16, 256)	1024
leaky_re_lu_2 (LeakyReLU)	(None, 16, 16, 256)	0
conv2d_transpose_2 (Conv2DTr	(None, 32, 32, 128)	294912
batch_normalization_3 (Batch	(None, 32, 32, 128)	512
leaky_re_lu_3 (LeakyReLU)	(None, 32, 32, 128)	0
conv2d_transpose_3 (Conv2DTr	(None, 64, 64, 64)	73728
batch_normalization_4 (Batch	(None, 64, 64, 64)	256
leaky_re_lu_4 (LeakyReLU)	(None, 64, 64, 64)	0
conv2d_transpose_4 (Conv2DTr	(None, 128, 128, 32)	18432
batch_normalization_5 (Batch	(None, 128, 128, 32)	128
leaky_re_lu_5 (LeakyReLU)	(None, 128, 128, 32)	0
conv2d_transpose_5 (Conv2DTr	(None, 256, 256, 3)	864

Total params: 7,994,080
Trainable params: 7,959,328
Non-trainable params: 34,752

a)

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 16)	448
leaky_re_lu_6 (LeakyReLU)	(None, 128, 128, 16)	0
dropout (Dropout)	(None, 128, 128, 16)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	4640
leaky_re_lu_7 (LeakyReLU)	(None, 64, 64, 32)	0
dropout_1 (Dropout)	(None, 64, 64, 32)	0
conv2d_2 (Conv2D)	(None, 32, 32, 64)	18496
leaky_re_lu_8 (LeakyReLU)	(None, 32, 32, 64)	0
dropout_2 (Dropout)	(None, 32, 32, 64)	0
conv2d_3 (Conv2D)	(None, 16, 16, 128)	73856
leaky_re_lu_9 (LeakyReLU)	(None, 16, 16, 128)	0
dropout_3 (Dropout)	(None, 16, 16, 128)	0
conv2d_4 (Conv2D)	(None, 8, 8, 512)	590336
leaky_re_lu_10 (LeakyReLU)	(None, 8, 8, 512)	0
dropout_4 (Dropout)	(None, 8, 8, 512)	0
conv2d_5 (Conv2D)	(None, 4, 4, 1024)	4719616
leaky_re_lu_11 (LeakyReLU)	(None, 4, 4, 1024)	0
dropout_5 (Dropout)	(None, 4, 4, 1024)	0
flatten (Flatten)	(None, 16384)	0
dense_1 (Dense)	(None, 1)	16385

Total params: 5,423,777
Trainable params: 5,423,777
Non-trainable params: 0

b)

Figure 4.9: Structure of each layer a) Generator b) Discriminator.

CHAPTER 5
EXPERIMENTS AND RESULTS

5.1 DATASET

Now, the visual datasets that we worked on in our study, the published ones and the training dataset required for the generative model are mentioned.

5.1.1. DAVIS: Densely Annotated Video Segmentation

The Densely Annotated Video Segmentation dataset (DAVIS) is a high quality and high resolution densely annotated video segmentation dataset under two resolutions, 480p and 1080p. It is publicly available. For this thesis, Bmx-bumps (Figure 5.4) and tennis (Figure 5.3) video series are used.

5.1.2. COCO (Common Objects in Context) Dataset

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has 1.5 million object instances, 80 object categories, 91 stuff categories, 1.5 million object instances. COCO has Detection, Keypoint, Panoptic, and DensePose categories to work. In this thesis, we used Figure (5.2,5.5,5.6) from COCO Dataset.

5.1.3. Training Dataset

It is not always easy to find the desired data in deep learning to train models. This can be a problem for most studies. In such cases, the desired learning was achieved by producing data with a single picture learning model [74] a result on Figure (5.1), which can be a method to solve the problem with the data we have. We approached the problem with a similar thought and created a training set by cutting sub-pictures from a single picture while working on the picture. We approached the

problem with a similar thought and created a training set by cutting sub-pictures from a single picture while working on the picture. We showed that the generative model trained with this dataset produces new pixels that can fill the gap created by the object to be deleted. With the same thought, we cropped sub-pictures over some previous and next frames to create a training dataset in video frames. It can be called this training set as "Crops Training Set (CTS)". Then it was tried to increase the number of training set by using the data augmentation methods such as angle, zoom and rotation.

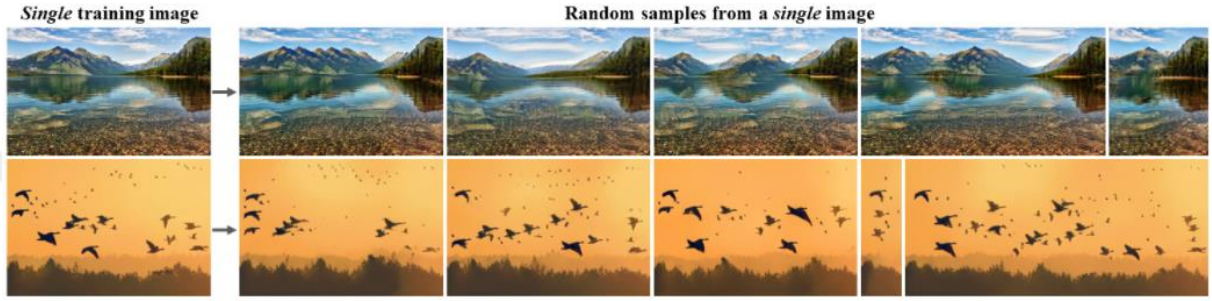


Figure 5.1: SinGAN can train a generative model from a single image, then generate random samples from the given image [74], we take a similar approach to create "Crops Training Set (CTS)".

5.2. IMPLEMENTATION DETAILS

3-channel RGB, (N, H, W,3) form (H: height, W: weight, N: number of images) images were used. While object detection is being pre-trained, the expected minimum is 224 pixel sets for the expected W, H value. Since the GAN network is designed to produce 256x256x3 images, the training set must be larger than that accordingly. As mentioned before, the training data set created on the visual data studied, by training with the GAN model, sub-images in the appropriate form were produced, in other words, new pixels were produced to replace the pixels defined as object location. The study was written in Windows 10 environment, pyhton programming language (3.7.9), Pytorch (1.6.0) [86] and Tensorflow (2.1.0) [84] libraries were used.

About the hardware used in this study:

- CPU: Intel(R) Core (TM) i5-10300H CPU @ 2.50GHz,
- X64, RAM 16,0 GB,
- GPU 0: Intel(R) UHD Graphics,

- GPU 1 NVIDIA GeForce GTX 1650 Ti.

While working some training time results in the below table:

Table 5.1 Time for generating new data, training different numbers of data and epoch

Time for epoch	Time	Number of cropped data training
2000	1 day, 4:10:05.025502	8364
5000	9:11:47.177279	472
5000	1 day, 2:46:08.546586	1488
5500	9:57:35.647046	461
10000	14:58:52.961629	389
10000	1 day, 1:52:51.305054	1350
10000	1 day, 6:05:50.494801	820
10001	7:49:42.184591	292
10000	7:06:26.215509	374

5.3. EXPERIMENTAL RESULTS

The result of our method by means of human visual perception is used as evaluation metric mostly. Because of that, some sample results as image Figure (5.2,5.5,5.6) are an image obtained from Coco 2017 dataset. Other results worked on Davis 2017 frame set dataset Figure (5.3,5,4) with acquired result as shown.

Figure 5.2: Sample image (640x480 jpg) from COCO 2017 Dataset, shadow is a problem.

Figure 5.3: Row a) A filmstrip set of frame samples named (tennis), b) Frame samples with objects removed, data from Davis (2017) Dataset (854x480 JPG format). Object location detected via Deeplabv3-ResNet101 for this series.

Implementation steps for each image/frame:

1. Image data generator => Crop and Save ROI s => Data augmentation
2. Generative Adversarial Network => Generate random samples
3. Object Detection => 'fcn_resnet101' / 'deeplabv3_resnet101'
4. Determine region to generate partial area on frame (with dilation) => Replacement from generated set to object location

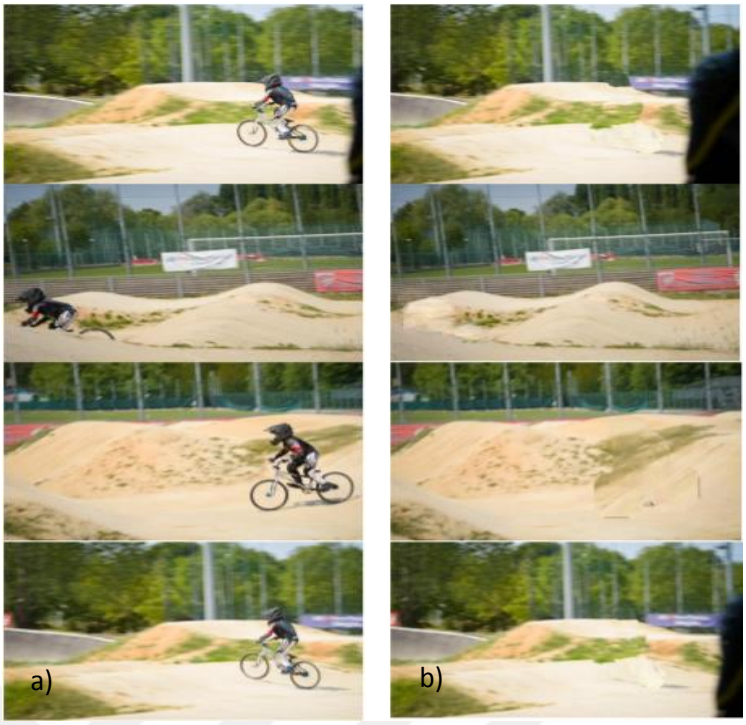


Figure 5.4: Column a) A set of frame samples (bmx-bumps), b) frame samples with objects removed, data from Davis (2017) dataset (854x480 JPG format).

Figure 5.5: A slightly more complex images with wire contents, a) column ours result b) column from [58].

Figure 5.6: a) Image (521x421 JPG) from COCO2017 Dataset, b) A result from [58] c) flower and dog not detected, people are removed, d) Umbrella not detected people, flower, and dog are removed.

For another evaluation criteria PSNR (Peak Signal-to-Noise Ratio) was used with results for below samples. PSNR is a method of calculating the difference between two images of the same size with some properties changed based on MSE (Mean Square Error) in equation (4), demonstrates in logarithmic decibel units. The smaller MSE means larger PSNR, larger PSNR means for better image quality. This method is calculated according to the equation (5). Accordingly, two similar images were compared, one is the image with the object removed and the other is the image that does not contain the object but has the same content.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{i=0}^{n-1}(O(i,j) - D(i,j))^2 \qquad (4)$$

$$PSNR = 10log_{10}(\frac{(L-1)^2}{MSE}) = 20log_{10}(\frac{L-1}{RMSE}) \qquad (5)$$

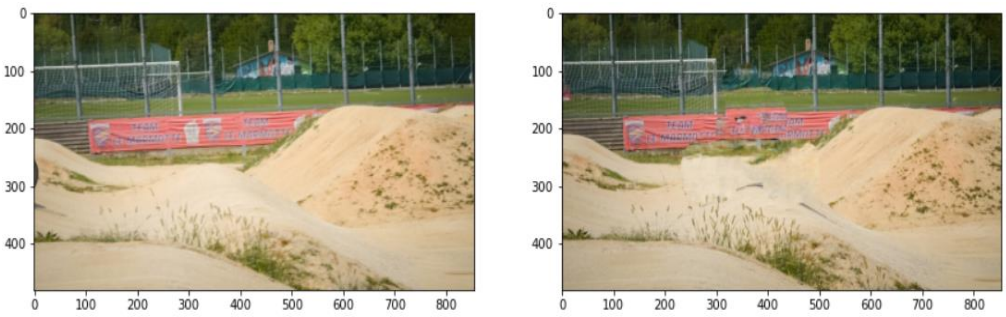


Figure 5.7 An image from DAVIS, bmx-bumps image (38) (left), removed object on image (22) (right).

The result for Figure 5.7: MSE: 85.56497771922977, PSNR: 28.807843192705263 dB.

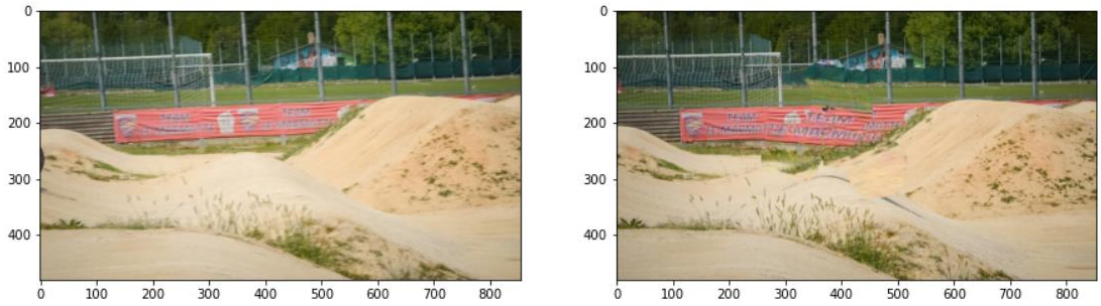


Figure 5.8 An image from DAVIS, bmx-bumps image (39) (left), removed object on image (21) (right).

The result for Figure 5.8: MSE: 87.45971571688784, PSNR: 28.712722995076838 Db.

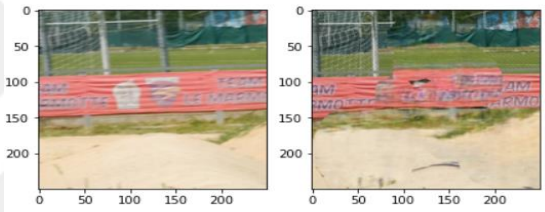


Figure 5.9 Evaluation on a selected ROI (left), same ROI remove object on it (right).

The result for Figure 5.9: MSE: 92.17016, PSNR: 28.484900194769892 dB.

CHAPTER 6
CONCLUSİON

6.1 CONCLUSION

This thesis was a study for removing the object from the object definition. It was a study that could contribute to the studies in this field by examining the studies on removing any object that we do not want to be in the frame/image. With this study, we can suggest the following. We realized that the most accurate determination of the set of pixels that make up the object class is a priority in producing the most accurate results. We have saved time by using pre-trained object-defined libraries that can detect quite quickly. Despite a long training process, we were able to produce new images that is effective to result with the GAN models. While we were searching for dataset to generate data which relocate of object, we saw that we can produce results from the data we have. We created a new training set for each input with the data set consisting of crop images. For the most accurate result, manually cropped images with similar content to the area of the object increased the quality of the produced content in creating an effective data set. At the same time, the detected object ROI (region of interested) should be included in the training dataset. While increasing the data, attention should be paid to the values used in operations such as zooming and rotating. Too large numbers degrade the quality of the produced picture. We have seen the effectiveness of the results produced by the deep convolutional generative adversarial networks model in the object deletion process.

6.2 FUTURE WORK

The training datasets used in this study were created manually. For this, while creating the most useful training dataset in future studies, automatic and effective datasets can be created with a learning-oriented data acquisition model. Learning and

production time takes quite a long time, to reduce this process, if the first randomly generated input matrix sent to the generating network is created from more meaningful and closer to the result data, the production process will be shortened. Using a meaningful input source instead of a random input will shorten the generation process.

REFERENCES

[1] M. WuDunn, J. Dunn, and A. Zakhor, *"Point Cloud Segmentation using RGB Drone Imagery,"* 2020 IEEE International Conference on Image Processing (ICIP), 2020, pp. 2750-2754, Doi: 10.1109/ICIP40778.2020.9191266.

[2] Serdar F. Tasel, Erkan U. Mumcuoglu, Reza Z. Hassanpour, Guy Perkins**,** *A validated active contour method driven by parabolic arc model for detection and segmentation of mitochondria, Journal of Structural Biology,* 10.1016/j.jsb.2016.03.002, 194, 3, (253-271), (2016).

[3] H. G. Kaganami and Z. Beiji, *"Region-Based Segmentation versus Edge Detection,"* 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, pp. 1217-1221, Doi: 10.1109/IIH-MSP.2009.13.

[4] Myilsamy, Radha, *Edge Detection Techniques for Image Segmentation.* International Journal of Computer Science and Information Technology, 2011, 3. 259-267.

[5] Kaur, Dilpreet and Yadwinder Kaur. *"Various Image Segmentation Techniques: A Review."* (2014).

[6] J. Sliž and J. Mikulka**,** *"Advanced image segmentation methods using partial differential equations: A concise comparison,"* 2016 Progress in Electromagnetic Research Symposium (PIERS), 2016, pp. 1809-1812, Doi: 10.1109/PIERS.2016.7734800.

[7] Xiaohua Tian and W. Yu, *"Color image segmentation based on watershed transform and feature clustering,"* 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1830-1833, Doi: 10.1109/IMCEC.2016.7867535.

[8] Dunn, J. C. (1973-01-01). *"A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters"*. Journal of Cybernetics. 3 (3): 32–57. doi:10.1080/01969727308546046. ISSN 0022-0280.

[9] Bezdek, James C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms.* ISBN 0-306-40671-3

[10] Y. Song and H. Yan, *"Image Segmentation Techniques Overview,"* 2017 Asia Modelling Symposium (AMS), 2017, pp. 103-107, Doi: 10.1109/AMS.2017.24.

[11] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, *"Densely Connected Convolutional Networks,"* 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269, Doi: 10.1109/CVPR.2017.243.

[12] R. Girshick, *"Fast R-CNN,"* 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448, Doi: 10.1109/ICCV.2015.169.

[13] S. Ren, K. He, R. Girshick and J. Sun**,** *"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi:10.1109/TPAMI.2016.2577031.

[14] Jifeng Dai, Yi Li, Kaiming He, Jian Sun. *R-FCN: Object Detection via Region-based Fully Convolutional Networks,* arXiv:1605.06409v2 [cs.CV] 21 Jun 2016.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg., *SSD: Single Shot MultiBox Detector*, arXiv:1512.02325v5 [cs.CV] 29 Dec 2016.

[16] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, *"Subcategory-Aware Convolutional Neural Networks for Object Proposals and Detection,"* 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 924-933, Doi: 10.1109/WACV.2017.108.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.*, arXiv:1606.00915v2 [cs.CV] 12 May 2017

[18] K. He, G. Gkioxari, P. Dollár and R. Girshick, *"Mask R-CNN,"* 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, Doi: 10.1109/ICCV.2017.322.

[19] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. *Fully convolutional instance-aware semantic segmentation.* In CVPR, 2017.

[20] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, *"OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172-186, 1 Jan. 2021, Doi: 10.1109/TPAMI.2019.2929257.

[21] H. Fang, S. Xie, Y. Tai and C. Lu, *"RMPE: Regional Multi-person Pose Estimation,"* 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2353-2362, Doi: 10.1109/ICCV.2017.256.

[22] S. Zhang et al., *"Pose2Seg: Detection Free Human Instance Segmentation,"* 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 889-898, Doi: 10.1109/CVPR.2019.00098.

[23] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, Piotr Dollár**,** *"Panoptic Segmentation"***,** arXiv:1801.00868v3 [cs.CV] 10 Apr 2019.

[24] X. Zhou, D. Wang, P. Krähenbühl, *"Objects as Points",* arXiv:1904.07850v2 [cs.CV] 25 Apr 2019.

[25] D. Guanlin, *"Research on Semantic Segmentation Algorithm Based on Deep Learning Control Tools,"* 2020 International Conference on Computer Communication and Network Security (CCNS), 2020, pp. 35-38, Doi: 10.1109/CCNS50731.2020.00016.

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, *"YOLACT++: Better Real-time Instance Segmentation,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, Doi: 10.1109/TPAMI.2020.3014297.

[27] J. Redmon, S. Divvala, R. Girshick and A. Farhadi**,** *"You Only Look Once: Unified, Real-Time Object Detection,"* 2016 IEEE Conference on Computer Vision and Pattern Recognition -(CVPR), 2016, pp. 779-788, Doi: 10.1109/CVPR.2016.91.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *"Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,"* 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587, Doi: 10.1109/CVPR.2014.81.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, arXiv:1406.4729v4 [cs.CV] 23 Apr 2015.

[30] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam, *"Rethinking Atrous Convolution for Semantic Image Segmentation",* arXiv:1706.05587v3 [cs.CV] 5 Dec 2017

[31] Ian J. Goodfellow, Jean Pouget-Abadie_, Mehdi Mirza, Bing Xu, David Warde-Farley,

Sherjil Ozairy, Aaron Courville, Yoshua Bengio. *Generative Adversarial Nets,* arXiv:1406.2661v1 [stat.ML] 10 Jun 2014

[32] Ian Goodfellow, *NIPS 2016 Tutorial: Generative Adversarial Networks,* arXiv:1701.00160v4 [cs. LG] 3 Apr 2017.

[33] Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu*, "Pixel Recurrent Neural Networks",* arXiv:1601.06759v3 [cs.CV] 19 Aug 2016.

[34] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, Koray Kavukcuoglu, *"Conditional Image Generation with PixelCNN Decoders",* arXiv:1606.05328v2 [cs.CV] 18 Jun 2016.

[35] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, *Progressive Growing of GANs for Improved Quality, Stability, and Variation,* arXiv:1710.10196v3 [cs.NE] 26 Feb 2018.

[36] Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros. *Image-to-Image Translation with Conditional Adversarial Networks.* arXiv:1611.07004v3 [cs.CV] 26 Nov 2018

[37] Zheng Xu, Michael Wilber, Chen Fang, Aaron Hertzmann, Hailin Jin, *Learning from Multi-domain Artistic Images for Arbitrary Style Transfer,* arXiv:1805.09987v2 [cs.CV] 14 Apr 2019.

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, Bryan Catanzar, *"High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs",* arXiv:1711.11585v2 [cs.CV] 20 Aug 2018.

[39] S. Yu, H. Dong, F. Liang, Y. Mo, C. Wu, and Y. Guo**,** *"SIMGAN: Photo-Realistic Semantic Image Manipulation Using Generative Adversarial Networks,"* 2019 IEEE

International Conference on Image Processing (ICIP), 2019, pp. 734-738, doi: 10.1109/ICIP.2019.8804285.

[40] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, *Semantic Photo Manipulation with a Generative Image Prior,* arXiv:2005.07727v2 [cs.CV] 12 Sep 2020.

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros**,** *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,* arXiv:1703.10593v7 [cs.CV] 24 Aug 2020.

[42] G. Antipov, M. Baccouche and J. Dugelay, *"Face aging with conditional generative adversarial networks,"* 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 2089-2093, doi: 10.1109/ICIP.2017.8296650.

[43] Huikai Wu, Shuai Zheng, Junge Zhang, Kaiqi Huang, *GP-GAN: Towards Realistic High-Resolution Image Blending,* arXiv:1703.07195v3 [cs.CV] 5 Aug 2019.

[44] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,* arXiv:1609.04802v5 [cs.CV] 25 May 2017.

[45] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, Pieter Abbeel, *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets,* arXiv:1606.03657v1 [cs. LG] 12 Jun 2016

[46] Alec Radford, Luke Metz, Soumith Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,* arXiv:1511.06434v2 [cs. LG] 7 Jan 2016.

[47] Martin Arjovsky, Soumith Chintala, Léon Bottou, *Wasserstein GAN,* arXiv:1701.07875v3 [stat.ML] 6 Dec 2017.

[48] D. Jansari, S. Parmar and G. Saha, *"Real-time object tracking using color-based probability matching,"* 2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC), 2013, pp. 1-6, Doi: 10.1109/ISPCC.2013.6663399.

[49] T. B. Suja and M. John, *"Fusion based object detection,"* 2010 National Conference on Communications (NCC), 2010, pp. 1-3, Doi: 10.1109/NCC.2010.5430203.

[50] Uijlings, Jasper & Sande, K. & Gevers, T. & Smeulders, A.W.M. (2013). *Selective Search for Object Recognition.* International Journal of Computer Vision. 104. 154-171. 10.1007/s11263-013-0620-5.

[51] F. Dornaika and F. Chakik, *"Efficient Object Detection and Matching Using Feature Classification,"* 2010 20th International Conference on Pattern Recognition, 2010, pp. 3073-3076, Doi: 10.1109/ICPR.2010.753.

[52] A. Criminisi, P. Perez, and K. Toyama, *"Region filling and object removal by exemplar-based image inpainting,"* in IEEE Transactions on Image Processing, vol. 13, no. 9, pp. 1200-1212, Sept. 2004, Doi: 10.1109/TIP.2004.833105.

[53] Y. Wexler, E. Shechtman and M. Irani, *"Space-Time Completion of Video,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 3, pp. 463-476, March 2007, Doi: 10.1109/TPAMI.2007.60.

[54] Z. Tang, Z. Miao, Y. Wan and J. Li**,** *"Automatic foreground extraction for images and videos"*,2010 IEEE International Conference on Image Processing, 2010, pp. 2993-2996, Doi: 10.1109/ICIP.2010.5649226.

[55] N. S. Sakpal and M. Sabnis, *"Adaptive Background Subtraction in Images,"* 2018 International Conference on Advances in Communication and Computing Technology (ICACCT), 2018, pp. 439-444, Doi: 10.1109/ICACCT.2018.8529323.

[56] C. Hima, M. Baburaj and S. N. George, *"A Novel Technique to Remove Marked Dynamic Object from Video Based on Reweighted Low Rank Tensor Completion,"* 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), 2018, pp. 25-29, Doi: 10.1109/CCCS.2018.8586802.

[57] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb, *Learning from Simulated and Unsupervised Images through Adversarial Training*, arXiv:1612.07828v2 [cs.CV] 19 Jul 2017.

[58] Rakshith Shetty, Mario Fritz, Bernt Schiele. *Adversarial Scene Editing: Automatic Object Removal from Weak Supervision.* arXiv:1806.01911v1 [cs.CV] 5 Jun 2018.

[59] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, Alexei A. Efros, *Context Encoders: Feature Learning by Inpainting*, arXiv:1604.07379v2 [cs.CV] 21 Nov 2016.

[60] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z. Qureshi, Mehran Ebrahimi. *Edge Connect: Generative Image Inpainting with Adversarial Edge Learning,* arXiv:1901.00212v3 [cs.CV] 11 Jan 2019.

[61] Yu, Jiahui & Lin, Zhe & Yang, Jimei & Shen, Xiaohui & Lu, Xin. *Free-Form Image Inpainting with Gated Convolution.* (2018).

[62] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. *"Globally and Locally Consistent Image Completion"*. ACM Transaction on Graphics (Proc. of SIGGRAPH), 2017.

[63] Alexandru Telea, *An Image Inpainting Technique Based on the Fast-Marching Method*, Journal of Graphics Tools, 2004, 9:1, 23-34, DOI: 10.1080/10867651.2004.10487596.

[64] T. Sun, W. Fang, W. Chen, Y. Yao, F. Bi, and B. Wu*, "High-Resolution Image Inpainting Based on Multi-Scale Neural Network," Electronics*, vol. 8, no. 11, p. 1370, Nov. 2019.

[65] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, Hao Li. *High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis,* arXiv:1611.09969v2 [cs.CV] 13 Apr 2017.

[66] Huang, Jia-Bin & Kang, Sing Bing & Ahuja, Narendra & Kopf, Johannes. (2014). *Image Completion using Planar Structure Guidance.* ACM Transactions on Graphics. 33. 1-10. 10.1145/2601097.2601205.

[67] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, Bryan Catanzaro. *Image Inpainting for Irregular Holes Using Partial Convolutions.* arXiv:1804.07723v2 [cs.CV] 15 Dec 2018

[68] A. Kaur, A. Raj, N. Jayanthi and S. Indu, *"Inpainting of Irregular Holes in a Manuscript using UNet and Partial Convolution,"* 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 778-784, Doi: 10.1109/ICIRCA48905.2020.9182917.

[69] H. Li and J. Huang**,** *"Localization of Deep Inpainting Using High-Pass Fully Convolutional Network,"* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 8300-8309, Doi: 10.1109/ICCV.2019.00839.

[70] Newson, Alasdair & Almansa, Andrés & Gousseau, Yann & Pérez, Patrick, *Non-Local Patch-Based Image Inpainting.* Image Processing On Line. 7. 373-385. 10.5201/ipol.2017.189, (2017).

[71] J. Pyo, Y. G. Rocha, A. Ghosh, K. Lee, G. In and T. Kuc, *"Object Removal and Inpainting from Image using Combined GANs,"* 2020 20th International Conference on Control, Automation and Systems (ICCAS), 2020, pp. 1116-1119, doi:10.23919/ICCAS50221.2020.9268330.

[72] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Do, *Semantic Image Inpainting with Deep Generative Models,* arXiv:1607.07539v3 [cs.CV] 13 Jul 2017.

[73] Emily Denton, Sam Gross, Rob Fergus, *Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks,* arXiv:1611.06430v1 [cs.CV] 19 Nov 2016

[74] Tamar Rott Shaham, Tali Dekel, Tomer Michaeli**.** *SinGAN: Learning a Generative Model from a Single Natural Image.* arXiv:1905.01164v2 [cs.CV] 4 Sep 2019.

[75] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li**,** *"Structure Flow: Image Inpainting via Structure-Aware Appearance Flow,"* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 181-190, Doi: 10.1109/ICCV.2019.00027.

[76] Soni, A., Pandey, N., & Halarnkar, P., *Review on Image Object Extraction,* International Journal of Current Engineering and Technology, E-ISSN 2277 – 4106, P-ISSN 2347 - 5161 (2014).

[77] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. *Flow-edge Guided Video Completion.* arXiv:2009.01835v1 [cs.CV] 3 Sep 2020.

[78] H. Zhang, L. Mai, H. Jin, Z. Wang, N. Xu, and J. Collomosse, *"An Internal Learning Approach to Video Inpainting,"* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 2720-2729, Doi: 10.1109/ICCV.2019.00281.

[79] S. Lee, S. W. Oh, D. Won and S. J. Kim, *"Copy-and-Paste Networks for Deep Video Inpainting,"* 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 4412-4420, Doi: 10.1109/ICCV.2019.00451.

[80] K. He, X. Zhang, S. Ren, and J. Sun, *"Deep Residual Learning for Image Recognition,"* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, Doi: 10.1109/CVPR.2016.90.

[81] Available: *https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/*

[82] Evan S., J. Long, T. Darrell, *Fully Convolutional Networks for Semantic Segmentation,* arXiv:1605.06211v1 [cs.CV] 20 May 2016.

[83] Available: https://pytorch.org/hub/pytorch_vision_fcn_resnet101/

[84] M. Abadi *et al., "TensorFlow: A system for large-scale machine learning."* 2016.

[85] Diederik P. Kingma and Jimmy Lei Ba., *Adam : A method for stochastic optimization.* 2014. arXiv:1412.6980v9

[86] Adam Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library", arXiv:1912.01703v1 [cs. LG] 3 Dec 2019

[87] K. Simonyan, A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition,* arXiv:1409.1556v6 [cs.CV] 10 Apr 2015

[88] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *"Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,"* 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587, Doi: 10.1109/CVPR.2014.81.

[89] P. Patel, A. Prajapati, S. Mishra, *Review of Different Inpainting Algorithms,* International Journal of Computer Applications (0975 – 8887) Volume 59– No.18, December 2012.

[90] M. Bertalmio, A. L. Bertozzi, G. Sapiro, *Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting,* IEEE CVPR, 2001.

[91] Y. Pritch, E. Kav-Venaki, and S. Peleg, *Shift-Map Image Editing,* IEEE ICCV'09, Kyoto, Sept. 2009.

[92] Available from: *https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73*

[93] Available from: *https://poloclub.github.io/ganlab/*

[94] Mehdi Mirza, Simon Osindero, *Conditional Generative Adversarial Nets,* arXiv:1411.1784v1 [cs. LG] 6 Nov 2014.

[95] N. Zhang and J. Fan, *"A lightweight object detection algorithm based on YOLOv3 for vehicle and pedestrian detection,"* 2021 IEEE Asia-Pacific Conference on Image

Processing, Electronics and Computers (IPEC), 2021, pp. 742-745, doi: 10.1109/IPEC51340.2021.9421214.

[96] P. K. Saha et al., *"Data Augmentation Technique to Expand Road Dataset Using Mask RCNN and Image Inpainting,"* 2021 International Conference on Intelligent Technologies (CONIT), 2021, pp. 1-6, doi: 10.1109/CONIT51480.2021.9498505.