Astana IT University

Yenglik Kadyr
Dayana Kassenova
Zhannur Murat

Segmentation of objects of interest in Video stream

5B070300 – Big Data Analysis

Diploma project

Supervisor
Aidarov K.
Associate professor

Republic of Kazakhstan
Nur-Sultan, 2022

CONTENTS

1 INTRODUCTION

1.1 Research Motivation

Nowadays, in the IT industry, especially in the sphere of AI, such as NLP, machine learning is developing intensively and rapidly. Machine learning is a part of artificial intelligence and has the peculiarity of not solving problems directly, but learning similar solutions. To further create and train such machine learning algorithms, several methods are used, such as mathematical statistics tools, probability theory, graph theory, mathematical analysis, and others.

At the moment, the utilization of AI technology in medicine is one of the most essential trends in the world. AI and neural networks can not only improve medical services, but also change, for example, diversify the diagnostic system, influence the emergence of new drugs, in a word, provide quality medicine and reduce costs. The Detectron2 libraries allow us to implement the intended program, for example, a program that analyzes skin videos and then detects skin cancer using that added video data.

Today, machine learning supports AI directions using Python libraries like detectron2, TensorFlow, and OpenCV that have been developed in a range of industries, from medicine to Smart City systems and government system automation. In medicine, the correctness of any action is the most important since a person's life directly depends on these same actions. In the modern world, with the development of medical technology, the utilization of artificial intelligence has facilitated the detection of diagnoses and diseases.

Relevance of the study: The distribution of objects according to geometric signs and characteristics such as color and size is one of the significant aspects of the action for detecting similarities and accurate recognition. Productivity is especially important for analyzing video data as video frames also, for using real-time recordings. The introduction of AI technology in the field of medicine is a significant and principally growing trend in the global healthcare system.

Theoretical significance of this project is to detect the probability of the accuracy of detection of skin cancer and the development of the rate of productivity. In addition, the practical part is appreciated by the fact that the creation of an apparatus in the form of a program with which doctors and oncologists can recognize the benignity or malignancy of cancer. This will certainly facilitate the work of doctors in the course of treating patients.

Practical significance: The system of segmentation and recognition of objects by geometric signs and characters is one of the most significant and relevant and is used in many areas, such as:

- Creation of topographic maps and GIS.
- Geological research (study of glaciers and snow cover, etc.);
- In the area of environmental protection,

- Medicine;
- Archaeological excavations;
- Planning and constructing a building or construction site.

1.2 Research Questions and Objectives

The goal of the work is to segment and identify the percentage significance of cancer using the deep learning from video data. With AI, or rather with the direction of machine learning, to implement a program that segments and refines the presence of a cancer risk into a percentage metric.

The main objectives to be fulfilled in the course of the study are:

1. Conduct analysis of existing popular object segmentation methods within videostream

2. Based on analysis, choose most appropriate for solving given the problem of segmentation

3. Implement chosen method of segmentation as a software component utilizing deep learning

4. Conduct experimental studies using implemented solution

5. Analyze obtained performance of segmentation based on established metrics/approaches

1.3 Research Methodology

Technological breakthroughs in the deployment of deep learning architecture in a variety of domains have already certainly contributed to the advancement of artificial intelligence technologies. As a result, deep learning is being used in a variety of industries for a variety of purposes, including financial services - accurately assessing credit risk, healthcare - natural language processing of handwritten notes, industry - modeling very comprehensive patterns, biometrics - recognizing human faces, and public - predicting highway conditions. One of the most important roles for deep learning in computer graphics technology is to solve real-world challenges. Deep learning may proficiently show experts with dependable and relevant analytical facts by locating the most important function. Furthermore, rather with machine learning, the process of training and learning models requires significantly more time and memory, as well as moderate and high-speed graphic cards and GPUs.

The Model divides a particular branch into two basic parts classification and coordinate analysis using semantic segmentation relying on R-CNN. The model delivers the original photo to the converter and separates it into two phases using the RPN (Region Proposal Network) process. The generated branches are joined to form the required configuration, and the result is shown via semantic segmentation. Mask R - CNN enhances the Faster R-CNN architecture by including a new branch that forecasts the position of the mask covering the

observed object, solving the instance segmentation problem. A mask is nothing more than a rectangular Matrix. Whereas 1 indicates that the Pixel is associated with the object, 0 shows that the Pixel is not associated with the item.

1.4 The Data Set

The dataset is intended to identify skin cancer, and we were permitted to utilize realistic video tapes in our research projects. Because video comprises a range of frames per second, we ended up with 174 600 photos. As a result, the Irfan View tool was used to sift high-quality photographs from existing ones, and a total of 1000 shots were chosen. To ensure appropriate dataset format, we installed labelme, a small visual editor that ran alongside Anaconda cmd.

1.5 Structure of the Project

The project consists of 2 significant parts such as Theoretical and Practical. First one contains the theoretical background information and related works with our methodology. Second one includes the current experiment of our methodology along with analysis and results.

1.6 Chapter 1 Summary

In summarization, for processing video data as video frames and using real-time recordings, performance is critical. The application of artificial intelligence technology in medicine is a significant and rapidly developing trend in the global health system. The theoretical relevance of this study is the establishment of a performance indicator and the accuracy of skin cancer diagnosis. Furthermore, the practical aspect is examined by the creation of an equipment in the form of a software that allows doctors and oncologists to detect cancer or malignancy. This, of course, makes doctors' jobs easier when it comes to treating patients.

2 Background and Literature Review

2.1 Theoretical background

Modern information systems and technologies produce data processing or support processes. The simplified procedure in clustering is considered to be the one that, according to the specified user criteria, provides data separation, and the most difficult one is the procedure that, according to the given description and nature, provides processes, visions, and approximate forecasts for determining the object. In the case of medicine, the recognition and finding of objects by any similarities, signs, and characters are still relevant to this day. In medicine, the term "photogrammetry"has the meaning of segmenting and detecting objects by their shape, size, and significant features. As a result, AI technologies, neural networks, and dataset analysis methods are widely used in medicine in the 21st century. Since AI provides not only high-quality and fast work, it also provides high-quality and highly accurate results.

AI in medicine - instead of human diligence, special customized algorithms and programs are used to analyze complex and voluminous medical data. Applications have been developed to monitor human health, where the relationship between treatments and patient treatments has been analyzed. There are a number of applications that are widely used in practice when prescribing treatments and prescriptions for diagnostics, responding quickly. AI in medicine is one of the main investment areas. Over the past half-century, there has been a sharp push and a big explosion in the development of medical technology. There are benefits associated with AI:

- Improving data productivity for faster collection and processing.

- Ensure the availability and volume of data that has been collected from medical devices of doctors and patients.

- Expanding the base of systematic genomic information.

If you believe the words in the journal Research and Markets were made preliminary, then by 2020 the AI market will have grown to $ 5.05 billion.

The selection of indicators facilitates the recognition or identification of objects. Unless determining the most relevant variables, both the qualities of objects and the capacity of the original picture waveforms to resolve must be considered. Let us look at how to process monochromatic (single-layer) photos to see that referring at. Each color can be treated separately using the methods explained in the color illustrations. Many geometric signs have the peculiarity of remaining unchanged when the object schema changes, and they come to invariance due to the normalization of geometric symbols relative to each other.

The classifying of objects is the foundation of automatic picture decoding. The pixel of a multiobjective picture belongs to a collection of spectral attribute values or a vector in dimensional space with a magnitude equal to the number of

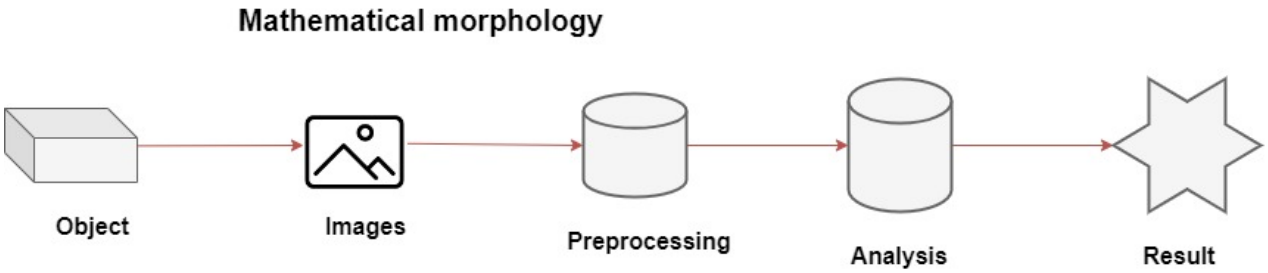

Figure 2.1 – Editing object images

viewing regions in this situation. The classification method is then shortened to sorting all of the grid components based on the reflection of each item, especially spatial color temperature in one or several spatial regions, such as agricultural crops or usage of land classifications.

First of all, difficulties in the segmentation of objects arise from variability. Most often, problems arise when the segmentation is unknown, since bitmap elements can sometimes interact and belong to several classes at the same time, which is why they are called mixed elements. On the other hand, in the process of recognizing the unknown, the unknown is ignored, and each element finds its place, its class. According to the rules of detection, it is usually combined and takes into account the features of objects that belong to a given class.

There are two kinds of data classification algorithms: supervised and unsupervised. The techniques of shifting from spatial brightness markers to entity subclasses are built on the training, testing, and referencing segment of the picture and afterward applied immediately to the remaining of the picture in supervised classifying. This kind of technique is called a trained algorithm.

Images with groups of similar (homogeneous) items are especially challenging to segment. Usually technologies would combine all of the items at one enormous cluster, but when these items overlay, it may be hard to identify whether it is single object or many. Recognition of objects by geometric features can be divided into two simple procedures, such as checking and calculating similarity. Calculate similarity is the received frames to compare with those already stored in the database, or in other words, the process to determine the identity. As a result, we get a similar picture to the one previously added by the user. It is this process that shows and proves the abundance of data in the database.

When checking a photo, the program, selecting a photo from the database, compares it with the added photo to identify similarities in the form of percentage significance. In simple terms, the program compares the added picture with the existing picture and shows the similarity percentage.

One of the most challenging aspects of object-oriented programming is establishing classes and objects. Even though the experience is demonstrated, this task frequently includes aspects of exploration and creation. Researchers may

now explore the fundamental aspects and operations that comprise the subject's language as a result of the findings. Researchers construct broad notions along with new processes that establish the laws on object interaction using innovation. As a consequence, exploration, and creativity are essential components of effective categorization. The goal of segmentation is to discover common qualities among items. Classifying items that have the same architecture or functionality into one category.

Photogrammetry is primarily a technical and scientific field, and concerned with estimating the form, size, placement, and kind of things in spacetime based on pictures of those items. Aime Lausedat, the inventor from France, noticed the implications of employing the freshly created camera in cartography in 1851, however, it took another 50 years for this technology to be effectively applied. Ground scanning, as it was subsequently defined, became extensively utilized in the period immediately before the First World War; all through the war, a far more successful technology of aero photogrammetry was adopted. Whereas aero photogrammetry was primarily employed for military activities until the conclusion of World War II, its field of use in peacetime has since broadened. Photo is presently the most common means of making images, especially in difficult-to-reach locations, but it is also commonly found in the analysis of biology and agriculture, along with other topics.

As is widely known, photogrammetry starts from the stage when a photograph appears and immediately proceeds to stereo photography. For example, the development of technology, flying vehicles, and optics has led to the rapid development of photometry and its use in many areas. Photogrammetry was carried out with cutting-edge equipment. Satellite photography, extraordinarily large photography, automated image scan, oversaturated colors photo, the use of media materials receptive to light outside the observable scope, and digital photogrammetry were all breakthroughs in the second half of the twentieth century. Photogrammetry employs techniques from a variety of fields, primarily lenses and projective geometries.

The space coordinates of an element's points are obtained in the basic instance by measuring two or more pictures collected from various places. In this scenario, similar spots within every picture are looked for. The visual stream is next collected from the camera's placement to the object's point. The alignment of a point in space is calculated of these rays. Further complicated programs can make advantage of already prior knowledge about an item, such as the symmetry of the features that compose it up, enabling them to recover the geographical coordinates of points just from one picture beneath particular circumstances. The algorithms that are used in photogrammetry are usually applied to reduce the sum of squares of errors, which are solved using the Levenberg-Marquard method

or the binding method, which is based on the least-squares method for solving nonlinear equations. When working with photogrammetry, the data type should adhere to such points as this diagram illustrates four types of data that have input and output when exposed to a photogrammetrist:

- the location of an object's points in space is determined by its spatial coordinates;

- the location of an object's points in an analog or digital image is determined by its photo coordinates.

- the camera's exterior orientation elements define its position in space and shooting direction;

- the geometric aspects of the shooting process are determined by internal orientation factors.

The easiest and most precise explanation of photogrammetry, which derives from the Greek language of this complicated term, is Measuring of lighting record. The preceding is a more comprehensive scholarly explanation. Photogrammetry is a branch of photography that enables people to gather snapshots using photo computing processes and unique capabilities, and then use those images to estimate the physical position and properties of actual things on Earth. Photogrammetry is also a novel remotely recommend method for identifying the geometric qualities of items and operations, evaluating them, and visually displaying information about a set of images collected from various camera angles.

The 3D coordinates of the projection Centroid, the transverse and longitudinal angle of inclination of the picture, and the rotation angle are all components of external perspective. The focal length of the lens, the kind of misrepresentations induced throughout photography can be acknowledged: for instance, lens deformities, displacement of material, and the 2D coordinates of the benchmark are among the elements of internal perspective. Clearly explained would aid in precisely assessing the distance and coordinates of an item's points, but also emphasizing the system's coordinate.

Computer Vision and Data Science are examples of AI approaches. They are distinguished by high-tech computational correctness, quality, efficiency, and well-known production credentials. Computer vision(CV) is an artificial intelligence branch that develops technological computers and systems capable of sensing, analyzing, and interpreting the visual environment by applying machine learning approaches. It detects similarities and extracts relevant information from digital photos, recordings, as well as other video data using ml algorithms. Humans are always bounded by apparently identical items that they come across daily. Even though they are both looking at and expressing the same issue, there are minor distinctions that differentiate them aside. The goal of computer vision is to detect and comprehend pictures in the same manner that people do, as well as to classify,

categorize, and organize them based on color and size.

Computer Vision Technology (CV): searches, monitors, classifies, and detects objects; extracts data from images and analyzes the data obtained.

- Shows items;
- Supports video analysts;
- Assists in the description of image and video content;
- Researches motion recognition and also handwriting technologies;
- Focuses on conceptual image processing.

Data Science is a specialized field concerned with finding trends in data in reasons to develop smart decisions, obtaining information from the data to the gathered object, and modeling them in a manner that can be handled by relevant stakeholders such as people, software and control devices. Data Science:

- Enhance knowledge;
- Find trends in information and forecast;
- Apply methodologies;
- Collaborate in fields like Econometrics, Statistics, Machine Learning, and Deep Learning.

Leading players inside the medical technology sector include Google, Apple, and Microsoft. Their artificial intelligence (AI) tools increase diagnosis accuracy, doctor availability, and medical data implementation. These large corporations have the advantage of having more resources and more talented workers. This enables them to design complicated goods with previously unavailable features. Google Health, for example, is a service that integrates many services for patients and doctors. It helps to avoid vision, breast cancer awareness, mental health, and other things with the use of AI.

Let us now discuss the importance of AI in each sector of medicine.

-AI in Surgery:

Even with years of practice, robotic technologies allow doctors with limited knowledge or other doctors with a specific surgical procedure to be treated at an unattainable level. The presence of the robot throughout the procedure reduces the influence of tremors on the attending physician's arm and eliminates unintentional movements. The Da Vinci surgical robot, which is regarded as one of the most advanced in the world, provides the doctor with a set of surgical instruments that may be used in minimally invasive surgery and increases control over common procedures.

- AI in diagnostics:

Images make up 90% of healthcare data, according to IBM, and their volume is growing faster than all other medical data. Neural networks, which were first used to recognize images of vehicles, dogs, and handwritten digits, have shown to be extremely useful in processing a wide range of visual input. AI-based

systems may evaluate medical photos and identify discovered traits, such as small tumors that the human eye may miss, after reading utilizing large data research. These algorithms recognize trends and provide information about the nature of departures from the norm, allowing doctors to save time.

The capabilities of neural networks are helping to alter the area of radiology, saving medical organizations time and money. The doctor should study the medical image obtained by MRI, computed tomography, ultrasound, or X-ray examination for any indicators of disease or anomalies. Several imaging studies must be clarified in order to identify any dangerous issue.If the patient has multiple photos taken over a period of time, artificial intelligence can understand the dynamics of the disease. As a result, Google conducted an experiment in which six certified radiologists were requested to examine the photographs in order to assess the AI-based system's effectiveness. Artificial intelligence performed as well as or better than humans in circumstances when the diagnosis was based on a single image. The technology correctly diagnosed 5% of cancer cases and cut false-positive sentences by 11

There are several trends in the development of AI, one of which is related to the integration of data types that are used for training. For example, for audiovisual speech recognition, a visual description of lip movement is integrated with an audio input to predict spoken words. Information that comes from different modal sources may have different predictive power and noise topology, while some sources may not have data. Heterogeneity of multimodal data makes it difficult to build models. It's critical to understand how to describe and summarize revenues in a way that reflects multiple modalities. Text, for example, is denoted by symbols, while audio and visual modalities are denoted by signals. All diagnostic information about a patient can be integrated into such multimodal data and processed by an AI system trained to consider the external image of a person and fragments of his body, as well as the results of analyses, MRI and CT scans, audio recordings of answers to questions, and so on. All of this enables us to create a universal diagnosis by taking a holistic approach to disease diagnosis and reducing the number of visits to various specialists for the appointment of efficient treatment approaches.

Health-related apps based on artificial intelligence

The potential of artificial intelligence to keep individuals alert means that they won't need to see a doctor. Artificial intelligence and the Internet of Medical Things (IoMT) are slowly shifting the healthcare paradigm from "reactive"to "active."Artificial intelligence and the Internet of Things will eventually combine to make linked gadgets intelligent for health monitoring. Diagnostics can be performed on a huge volume of data generated by AI or IoT.

2.2 Literature Review

The first research aimed to examine issues of both tracking and segmenting objects to reach a high-level operational efficiency. The main method to tackle the problem is using convolutional Siamese networks on Res-Net-50 architecture, where SiamRPN maximizes SiamFC's results by determining the target position using a bounding box of variable pixel density. Thus, box estimations are generated in tandem with classifying scores. The trained model even on 3 different datasets, such as Youtube-VOS, COCO, and ImageNet-vid, was better at first. Eventually, a mask-branched SiamMask model demonstrated the greatest mAP - IOU metrics with 71.68 % rather than others. However, it fails sometimes at a "non-object"trend and blurred action, maybe it happens due to a deficit of instances with identical characteristics in a training dataset. The predominant privileges of this approach were running online without adapting to testing with the highest speed at 60 frames per second [1].

In the next paper, the authors discuss pre-trained CNN, exactly Complementary CNN for conducting both foreground and background assessments of each frame. To predict precisely, they further partition each frame into a group of superpixels and create neighborhood bidirectional flow in frames. As a result, the key elements in different frames can appear gradually, while various sorts of interruptions can be avoided. From pictures, it can be observed that maps with and without backgrounds may depict definite items very clearly, but still, some mistakes are shown. Unfortunately, when it refers to original things, regions that are identical to the backdrop, shadow, and projections can generate wrong predictions. Thorough testing with three sets of video materials has proven that the described method, CCNN, is successful [2].

In this work, professors investigate again two challenges such as deep learning of the saliency model with a lack of pixelated annotated video data and rapid learning and detection of instances. For this reason, they announced a new algorithm, a fully convolutional network (FCN) for evaluating pixels. According to observations, it can be found that this video saliency model is both flexible and cost-effective while avoiding the computationally intensive optical flow calculation. Moreover, in comparison with activity recognition, video saliency may be derived from a relatively quick study of video sequences. While training, FBMS and Davis datasets with full annotation including 50 video sets were used. Mentioning results, the f-measure was 0.3, which is quite low performance, nevertheless, the time consumed was 0.47 seconds and only 2 frames per second on Nvidia G-force Titan X GPU. Regarding the matching of presence with the surroundings, this technique is used to extract both static and dynamic visual relevant data and accurately recognizes noticeable moving objects. Summing up, this approach is extremely quick, producing a synthetic sequence of video frames, an optical stream, and

pixel-by-pixel annotations all at the same time [3].

From this report, it is known that respondents created a new method for tracking vehicles on the roads during traffic monitoring issues. An invented technique derived from the Fast Region-based Convolutional Network (Fast R-CNN), that captures the complete picture and a series of pictures for the input, then generates bounding box placements with prediction depending on feature categories for the outcome. Obviously, it can be assumed that because of the superior performance of Fast R-CNN, it has a greater detection result than any earlier approaches. In such a manner, when other mechanisms are unable to identify automobiles properly, their novel method performs confidently on congested traffic roads. Specifically, the model trained on the PASCAL VOC 2012 dataset and reached impressive object detecting accuracy of 98%. Furthermore, the running time was 3 fps for counting transport on a GPU, as a result, obtained the amount of cars from the video into the database immediately. In general, this investigation operates quite well even with complicated recordings with overlaps or dense traffic reveal [4].

In this research paper, the authors aimed to handle a complex mission of segmenting multiple objects, particularly for the zero-shot situation, when unsupervised classification is defined as no need to initialize frames in video sets. As a solution, the proposed algorithm is a Recurrent neural network (RNN), which helps to encode the spatial and temporal change of video sequence items and also is similar to CNN, but modifies local networks by sharing parameters. Indeed, contributions of this approach can be noted such as adaptation for both zero and one-shot scenarios, and may not require processing after prediction. Henceforth, this model outperforms existing technologies in terms of analysis running time, hitting 44 ms per frame using P-100 GPU. Admittedly, it surpasses preceding VOS algorithms and gives excellent results with no need for fine adjustment for every test sequence, finally making it the quickest technique [5].

In this work, researchers explored how to eliminate the difficulty of frontal object recognition in the live stream. They recommended the architecture of autonomous video frame handling utilizing a deep generative adversarial network (GAN) that is capable of managing temporal consistency throughout frames while ensuring significant scores without the need for any direct trajectory relied upon knowledge. Important to mention that after adding the PDS loss function to the GAN, for the purpose of reducing the amount of wrongly predicted pixels during segmentation, which enhances the quantitative findings while also maintaining the training. Therefore, evaluation of this algorithm indicated 77.8 % impressive scores for region similarity, J-mean, on a Youtube-objects dataset. Even though, it works better with semi-supervised learning in contrast to unsupervised ones. Undoubtedly, their introduced solution behaves substantially

better while segmenting a single item of interest in challenging circumstances such as background interference, switching the focus, blurring of movement, overlaps, and shape displacements of things [6].

In this research, the authors presented a new method called YOLO, which is incredibly accurate and fast recognizes objects in real-time. The process starts by splitting one image by a grid, then getting each n bounding boxes in each grid in order to produce class probabilities for them. As a result, a Single Convolutional Neural Network forecasts an object with 45 fps more accurately as much it is possible computing on Titan X GPU. The prediction takes a 63.4 percent accuracy for the Pascal VOC dataset. The main drawbacks were improper localisations, and difficulty dealing with little items that occur in groupings. However, YOLO forecasts with a single network estimate, as opposed to systems like R-CNN, which require thousands of estimates for a single picture. Besides, if it is linked to a real-time camera, it operates as a track system, which recognizes things while they shift and alter their appearance. In summarising, YOLO detects objects efficiently and precisely, making it excellent for computer vision performs [7].

We presented a new paradigm for leveraging virtual reality to successfully increase existing data patterns in this study. For aware semantic segmentation, we train a contemporary Multitasking Network Cascade (MNC). [8]. The topic of this article was Deep Learning for Medical Image Segmentation. The segmentation of medical images was demonstrated using a variety of methods. Visualization methods like attention and class-activation-map are currently dominating medical image analysis interpretation (CAM). As a result, research into the interpretability of deep learning for medical picture segmentation will be a popular topic in the future [9].Deep learning-based optimization techniques for medical picture segmentation. Because medical images are typically 3D or 4D, CNN methods necessitate a high number of parameters as well as a significant amount of computer resources for training [10].

As additional DL-based techniques are developed, the limitations of DL in the identification and segmentation of medical centers are becoming increasingly apparent. By integrating the strong decision-making skill of RL with the high awareness of DL, DRL has an intentional process of attention focus that gradually creates evidence of trust in the detection of the item of interest. As a result, scientists are working to create advances in this sector in order to improve outcomes. DRL, a modern artificial intelligence method, works on a fundamentally different premise than DL and has proved its abilities in a range of other medical image analysis fields, like medical object recognition [11]. There were approximately a hundred approaches evaluated, with the following methods standing out: CT and CN, extended CNN, attention-based models, RN, R-CNN, generative and adversarial models, and others. We have compiled a quantitative

analysis of these models' performance on a variety of common tests, including PASCAL VOC, MS COCO, Citades, and COLLECTED 20 thousand data sets. [12]. Deep persuasion networks (DBN), which are suggested by a type of RDBMS, were utilized to segment the 2D instance in this study, and the R-CNN mask has been used to segment the 2D instance. DBN connected it to the active contour to achieve improved performance with a limited amount of training data. Despite their similar appearance, the suggested approach utilized two distinct DBNs for the segmentation of endocardial and epicardial pictures and obtained good accuracy. Mask R-CNN can recognize objects in an image and create a segmentation mask for every occurrence at the same time. Three biomedical data sets were used to test the chosen method: HL 60 cell nuclei, microglia cells, and developing C embryos [13]. In this article, we discussed how deep learning can be used to segment medical images. Data improvement, such as geometric transformation and color space improvement, is among the best alternatives to the problem of insufficient training data. GAN synthesizes new data using the old data. Another method for studying the segmentation of medical images in a limited sample is based on a meta-learning model [14].

We give a taxonomy of existing approaches for segmentation with partial control that allow the use of labeled data, unlabeled data, and previous knowledge. We considered segmentation with partially annotated areas, point annotations or partially annotated slices, interactive segmentation, and multiclass segmentation from several datasets branded with a partial class, and showed the current technical state of the latest solutions for the task of segmentation with partial control [15]. The most popular image segmentation methods are used for MRI segmentation. Computational efficiency will be especially important in real-time data processing applications such as computer-controlled surgery. New methods are usually developed to obtain more accurate results by including 3D neighborhood information and preliminary information from atlases. As a result, the segmentation process often becomes more complex and time-consuming [16].

The UNET neural network has been used in experiments to solve the problem of segmenting medical images. The feasibility of employing this technique for semantic picture processing has been demonstrated by experimental results. Other findings suggest that the UNET network learns faster and collaborates effectively with the Adam optimizer, but that modifying the loss function has minimal impact on the outcomes [17]. A color picture segmentation technique based on clustering in the space of the image's primary component has been created. The Pitas and Umbraugh algorithms were employed in this study. The original color image is converted into the main component space using this approach. When compared to segmentation algorithms that use clustering in the RGB color space, this simplifies the identification of clusters corresponding to a homogeneous area of the original

image, resulting in superior image segmentation [18].

Convolutional neural networks are the best choice for segmentation problems (CNN). Multiple libraries exist that allow you to use pre-built neural networks with many layers that have been trained on millions of photos. We created a 4-layer neural network utilizing the Unet architecture using the Karas framework. Each pixel in the original image can be an object of interest, and the input is an image in the form of a NumPy array. The output is a matrix [19]. The concepts "binarization,threshold,"and "segmentation"are all defined in this article. The image binarization process is classified straightforwardly. The Binarization and Segmentation module of the Vision Development Module is appropriate for LabVIEW program developers [20].

2.3 Chapter 2 Summary

The following conclusion may be reached from the preceding discussion: the goal of computer vision is to detect and comprehend pictures in the same manner that people do, as well as to classify, categorize, and organize them based on color and size. Moreover, artificial intelligence is integrated in medicine, particularly in surgery, diagnostics, MRI imaging, cardiograms,ultrasound findings, and so on. From literature review, we can assume that segmentation in a variety of spheres of medicine is quite actual issue, which solved using different methods such as Deep learning models, Canny and Laplacian techniques.

3 Data Overview

3.1 Data Collection

Dataset is aimed for detecting a skin cancer, and real video materials were allowed to use in our research studies.

As video contains from a variety of frames per second, we got from the doctor 10 videos with the length of about 15 minutes, in general 150 minutes, which is 9000 seconds with 20 frames, and finally we have got 174 600 images.

Consequently, the Irfan View program was utilized for sorting a high-quality images from existing, and in overall there selected 1000 photos.

For proper dataset format, we installed a little program as a graphic editor, named labelme, where Anaconda CMD was used to run. Labelme is a graphical interface software for labeling photos and used as a graphical editor for classifying data and producing machine learning datasets. After we have installed the required image, we will draw it with the mouse and mark one region as benign and the other as malignant. We must only focus on one zone at a time.

After each region has been labeled with its own category, the file is given a name. The category name and digit numbering must be included in the file name.

The file's json format will display automatically even though it has been saved. The two files that are present in the train directory are entered.

The software's database is separated into two groups. The Category2 reference can be viewed using the data path.

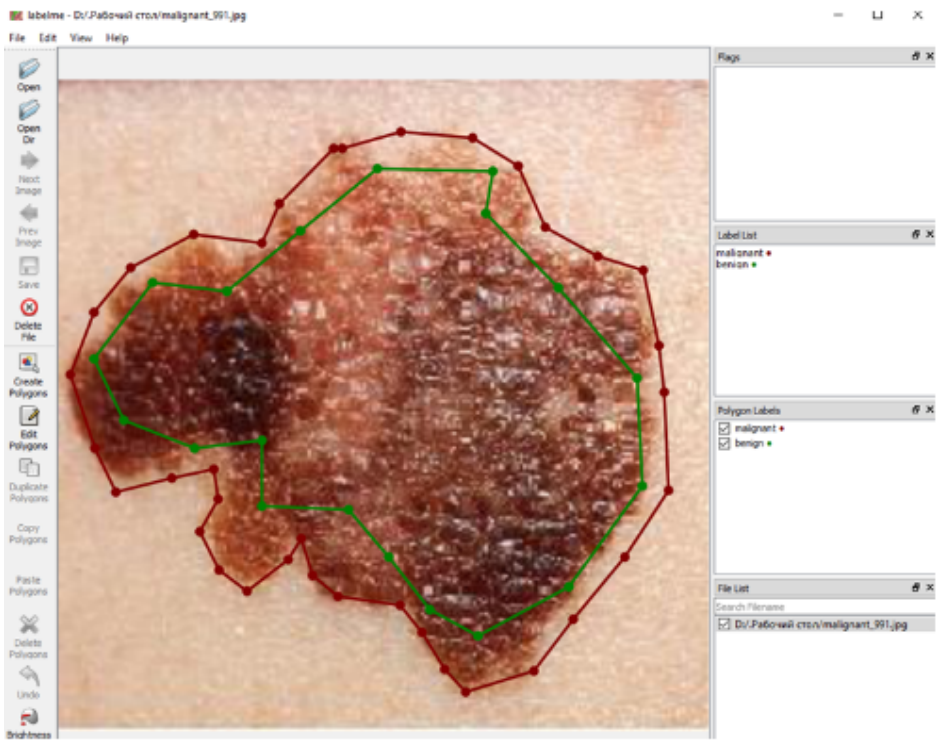

Figure 3.1 – labelme interface program.

Labelme's Operations step by step:

- The photo is inserted
- A special pencil is used to mark the location of items in the photo
- Once the object has been properly marked, it is divided into the chosen classification
- After classification, a hash - json file appears:

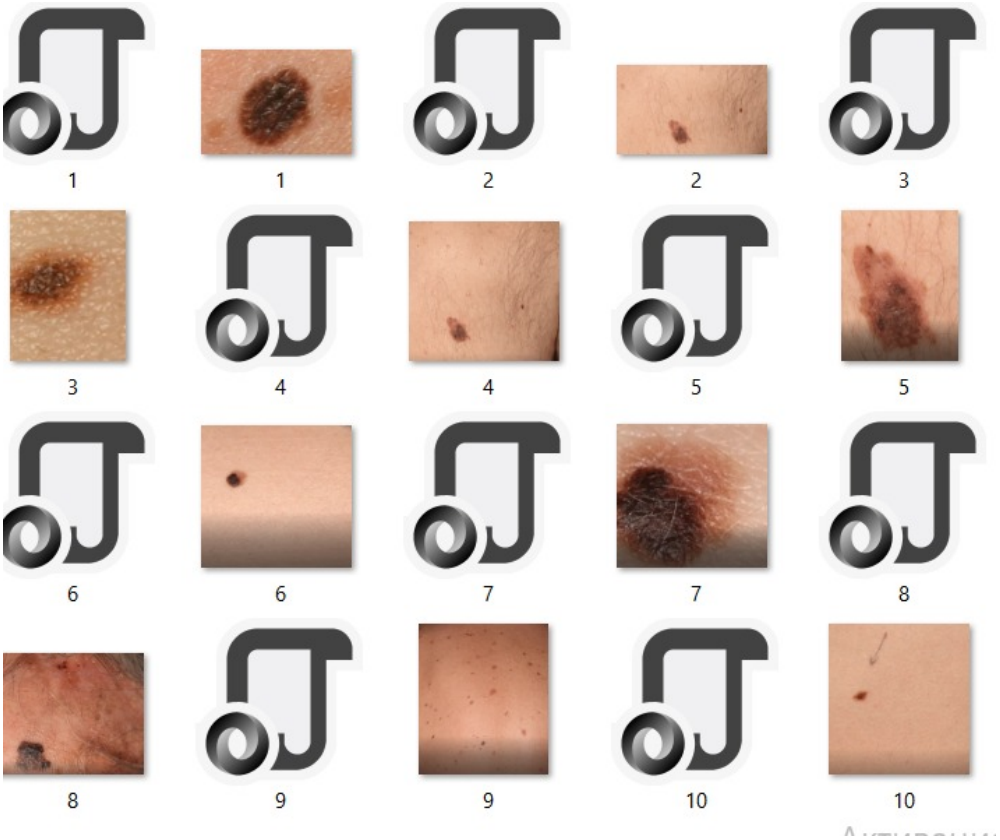

Figure 3.2 – Dataset

3.2 Data Summary

For the program to work well, it is necessary to take into account several points in the implementation of the program. For example, in the first place, the files that are downloaded frames should be of the size 224 * 224 because it is considered the most standard and principled size for further work.

By the help of using the labelme program, which is intended for recognition, should be divided into regions to identify a specific object. Downloading the image from the database was done with the help of this editor. As a result, we practically automatically receive a JSON file with hashed data where the data is stored in the image area.

ImageData is a JSON hash, and ImagePath is really the title of such a file to which JSON is associated. The relevant images are then sent to a different directory, where the software matches them to the database photos. Generally,

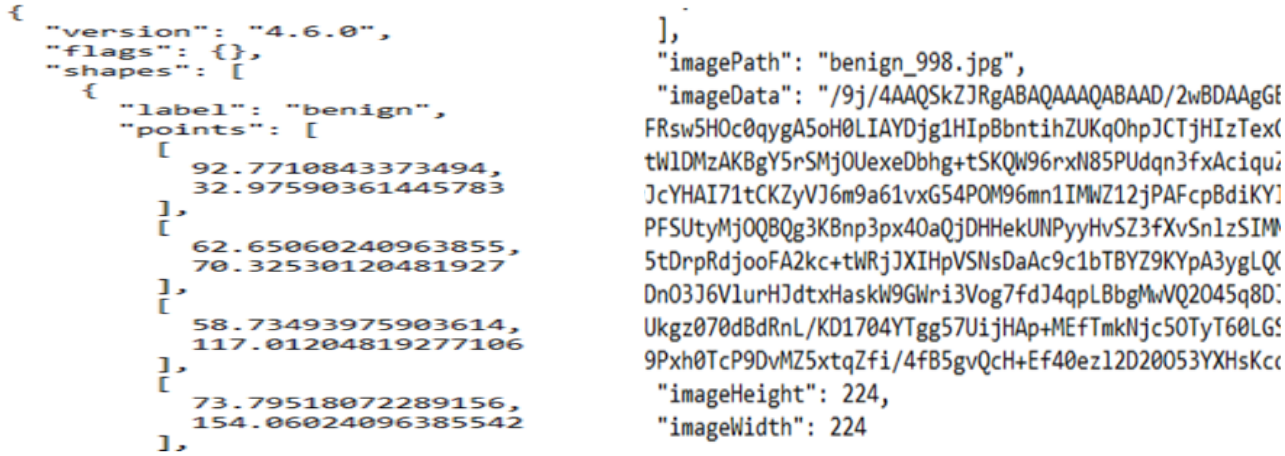

Figure 3.3 – layout of a JSON file in a text file.

our dataset contains these constitutional texts JavaScript Object Notation. A JSON file must be present and assigned to each frame. By studying this format, the program will be able not only to view the image but also to analyze all other parameters.

3.3 Exploratory Data Analysis

Data visualization and analysis are conducted to better comprehend the data trends. The pie chart below depicts the percentages of used and selected images from collected all pictures from video materials:

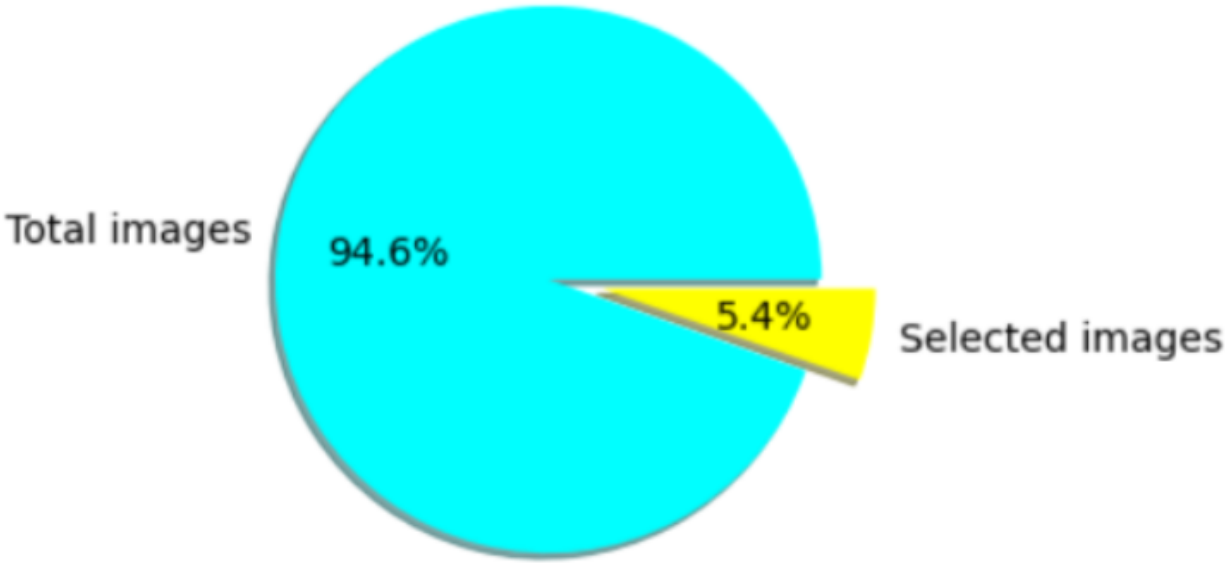

Figure 3.4 – Pie chart of used images

The accompanying line graph shows that red color has some high value spots in the real image, which was taken from dataset:

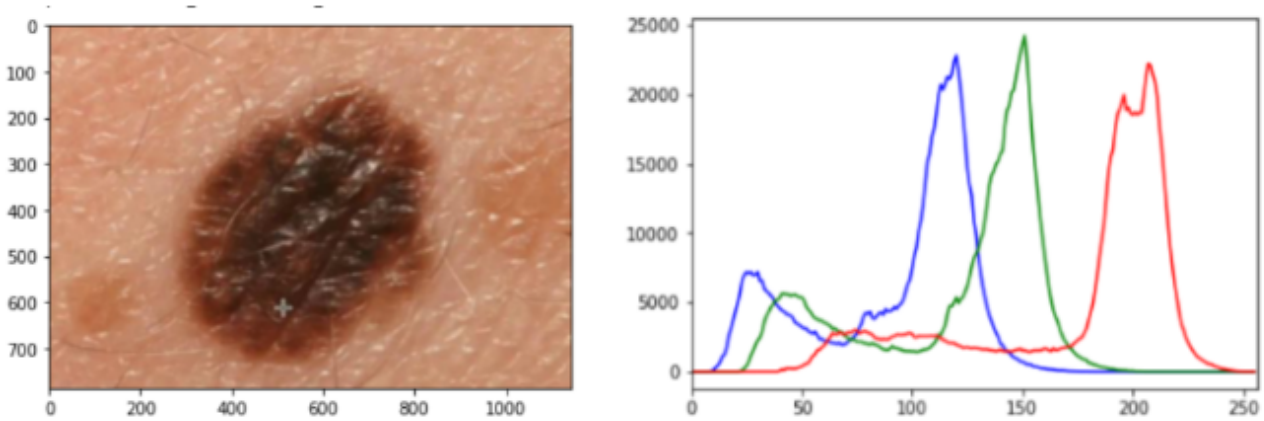

Figure 3.5 – RGB version of the image

3.4 Chapter 3 Summary

To sum up, labelme software, which is designed for recognition, should be separated into regions in order to identify a certain item. The image was downloaded from the database with the assistance of this editor. As a result, we almost always receive a JSON file containing hashed data that is saved in the picture area. Some exploratory data analysis of total and selected images is done.

4 Methodology

The action of segmenting a digital photo into many pieces is known as segmentation. The goal of segmentation is to render a picture accessible to analyze by clarifying or transforming its appearance. Image segmentation is a technique for separating objects and boundaries, lines, curves, and so on in frames. Segmentation, to put this into perspective, is the method of allocating certain important qualities to each pixel of an image related to visual features of pixels with much the same features. The distinctions between the three parts are depicted in the diagram below. As you can see, instance segmentation is compatible with a wide range of compositions:

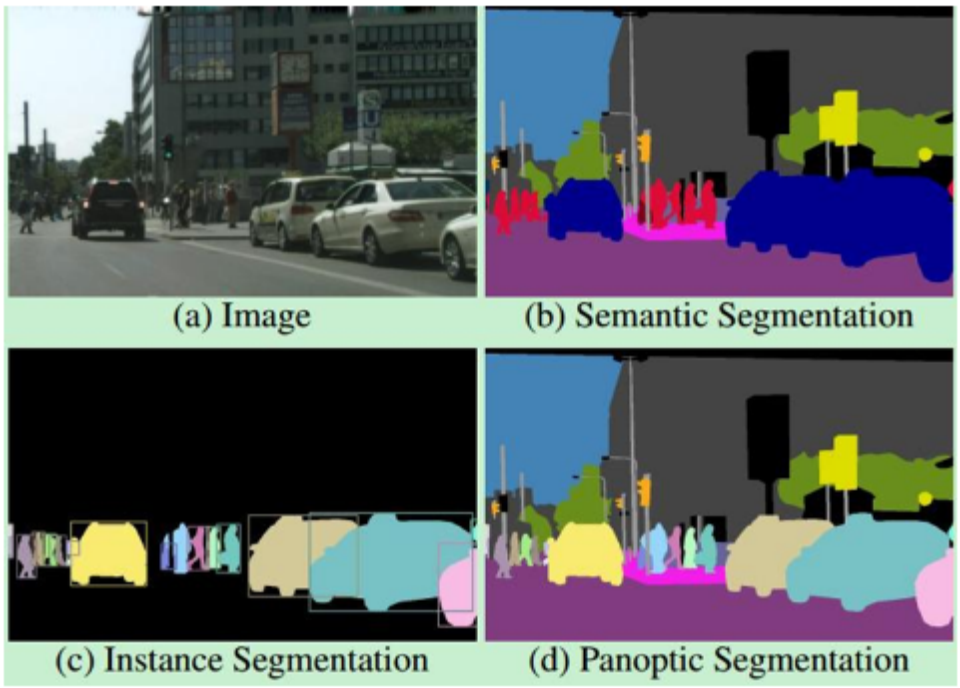

Figure 4.1 – Segmentation types

- Semantic segmentation: classifies a pixel in a picture after another.
- Panoptic segmentation: represents all of the items in an image.
- Instance segmentation: recognizes single object and segments items in a picture.

Deep learning(DL) is a profound branch of machine learning. Traditional machine learning(ML) is the process of extracting new information from large amounts of data going through the mechanism. Deep learning networks involve enormous volumes of unlabeled data to form proper inferences, whereas ML can work with modest quantities of data supplied by individuals. Along it demands people to precisely specify all procedures and also can make their own decisions

with sufficient and clear integrity, meanwhile, DL autonomously develops new procedures. By finding the most significant function, deep learning can proficiently present professionals with reliable and relevant analytical data. In addition, the process of training and learning models takes considerably longer time, and a substantial weight of memory and necessitates moderate and high-speed graphic cards along with GPUs rather than machine learning.

Technological breakthroughs in the deployment of deep learning architecture in a variety of domains have already certainly contributed to the advancement of artificial intelligence technologies. The word "deep"in the phrase "deep learning"corresponds to the amount of layers that a neural network accumulates through time, during which the deeper the network develops, subsequently, the higher will be performance. The network's input data is processed in a specific method by each layer, which then updates the following layer. As a result, one level's output becomes the input for the subsequent. As the system improves its model, deep learning network preparation requires a while and demands the collection and processing of vast volumes of data during tests. However, the rapid flow of incoming data gives limited time for a successful development procedure. As a result, experts must modify deep learning algorithms so that neural networks can analyze massive amounts of ongoing input information.

Furthermore, neural networks have been presented since the 1950s, and programming power and data warehouse competencies have lately advanced to the point where deep learning techniques may be employed to generate incredible technological innovations[21].Besides, DL is made up of many layers of artificial neural networks, and it employs nonlinear operations and generalizes model representations in huge datasets. In general, the basic principle of deep implementation involves the system building as much of its own functionality as practicable. It may be used to optimize outcomes and reduce computation time in a variety of digital operations.

Accordingly, DL is deployed in a multitude of sectors for a variety of purposes such as financial services - accurately assessing credit risk, healthcare - natural language processing of handwritten notes, industry - modeling very comprehensive patterns, biometrics - recognizing human faces, and public - predicting the condition on the highways. Especially, amongst the most prominent responsibilities for deep learning in computer visualization technology is solving real-world problems.

MASK R-CNN architecture

The Python programming language is utilized in artificial intelligence systems to implement the Mask R-CNN neural architecture. By specifying the required layout, the Mask R-CNN neural architecture operates on the accepting parameters idea. The following is the functioning proposition of the Mask R-CNN model:

- Divides objects into pictures using the matching occurrences of classes with the relevant information.

- Objects are categorized by the user according to a set of criteria.

- Issues a precise message to the user for the specified items, followed by some settings.

At first, Mask R-CNN is fundamentally based on a Faster R-CNN extension by including a second branch that forecasts the position of the mask supposed to cover the observed object, resolving the instance segmentation problem. A mask is simply a rectangular Grid. Where 1 denotes if the Pixel is part of the object and 0 denotes that the Pixel is not part of the object.

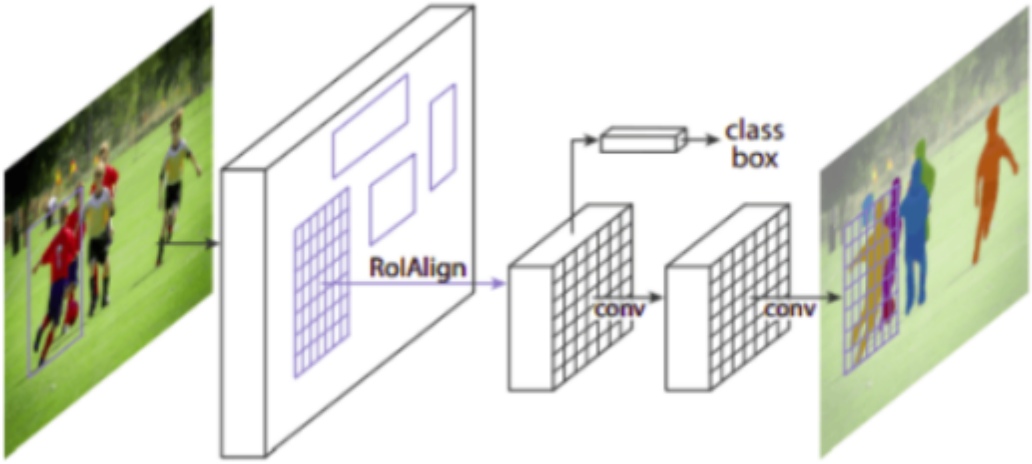

Figure 4.2 – Mask R-CNN process

In addition, faster R-CNN is extensively utilized for object detection applications. It produces the target class and bounding box dimensions on every item in an image for a given picture. Furthermore, Mask R-CNN neural topology was limited to categorizing objects based on a single feature. Most elements of the Mask R-CNN model are now focused to segmentation and display.

Let's take a relatively brief look towards how Faster R-CNN performs. It will also help understand the logic underlying Mask R-CNN:

1 Faster R-CNN starts by extracting keypoint maps from the pics employing a ConvNet.

2 The potential bounding boxes are subsequently supplied after passing those feature maps via a Region Proposal Network (RPN).

3 The RoI pooling layer is then implemented to these bounding boxes, putting all of the options to almost the same size.

4 Ultimately, the ideas are sent to a fully connected layer, which classifies and produces object bounding boxes.

Understanding Mask R-CNN will be simple after learning how Faster R-CNN operates. The established architecture was logically separated between CNN-a network for evaluating image characteristics, which they name the backbone, and head — an organization of aimed to elucidate for forecasting the surround frame, identifying the item, and establishing its mask. They all share the Loss function, which has three components:

$$L = L_{class} + + L_{box} + L_{mask}$$

Afterwards, let us concentrate into the Mask R-CNN operating structure, beginning from input and working our way to guessing the target class, also bounding box, and finally an object mask:

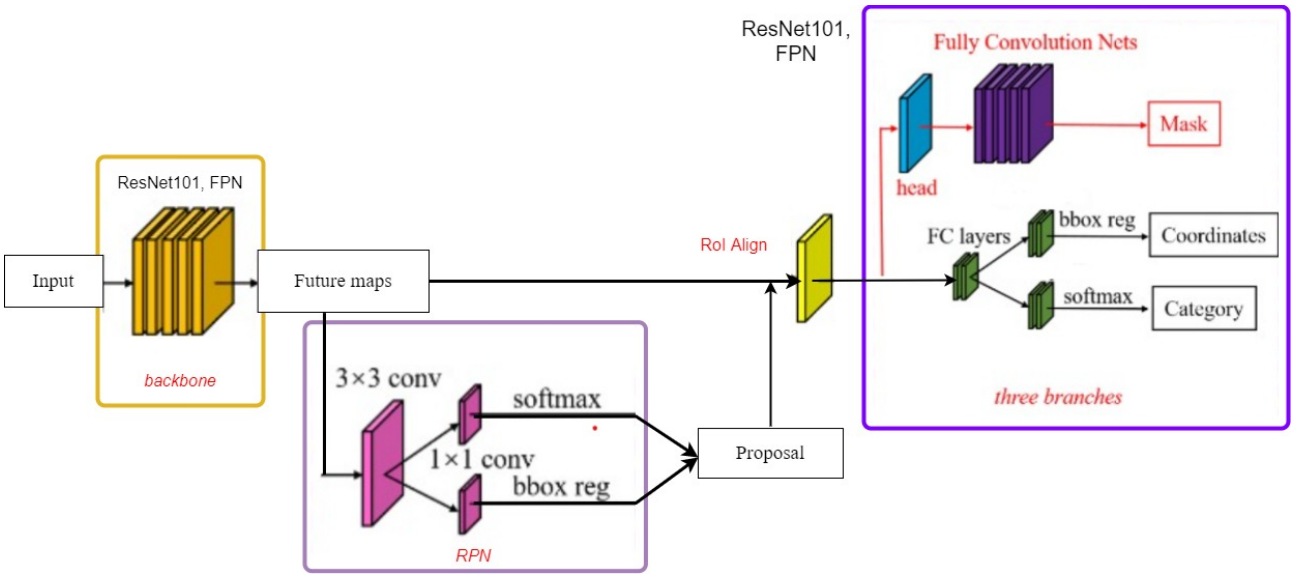

Figure 4.3 – MASK R-CNN architecture

Model of the Backbone

In Mask R-CNN, the first task is apply the ResNet-101 architecture for deriving features from the pictures, just like how we use the ConvNet in Faster R-CNN to extract feature maps from the image. Such characteristics serve as an input to the following layer in Figure 4.4.

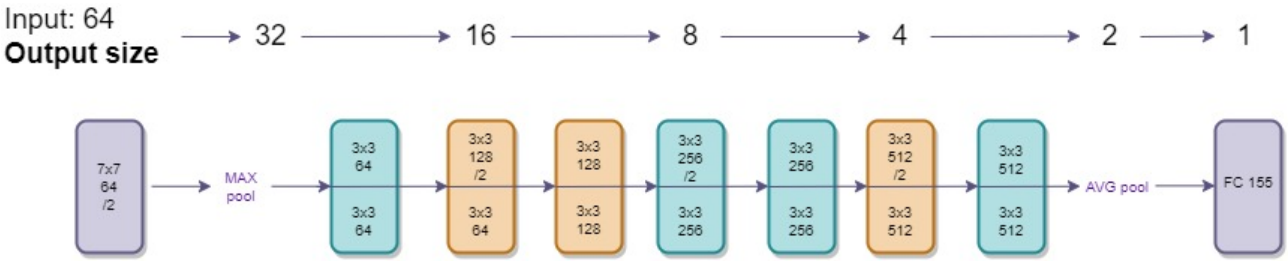

Figure 4.4 – ResNet-101 architecture

Region Proposal Network (RPN)

We subsequently attach a region proposal network to those feature maps acquired in the first phase. This simply forecasts whether or not an object is existent in that area in Figure 4.5.

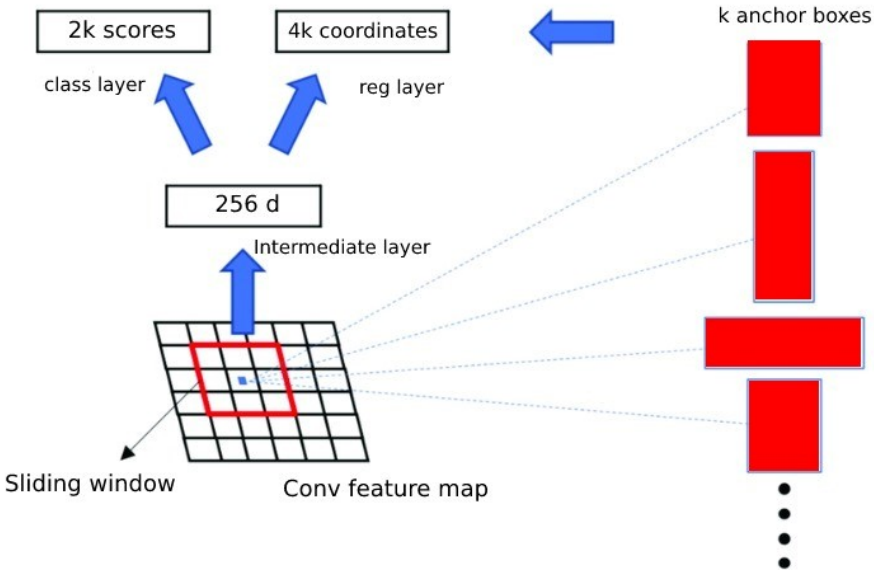

Figure 4.5 – RPN structure

Region of Interest (RoI)

As an outcome, we have been using a pooling layer to transform all of the areas into this kind of shape. The target class and bounding box are then projected by running these areas throughout a fully connected network. Moreover, these procedures are nearly identical to how Faster R-CNN functions up to this moment. The distinction between the two frameworks immediately arises. The mask is likewise constructed in Mask R-CNN

Segmentation Mask

We may add a mask branch to the existing architecture once we get the RoIs depending on the Intersection over Union values. This function gives the mask for each zone that includes an item. It generates a mask of size 28 X 28 for each zone, which is subsequently amplified for inference.

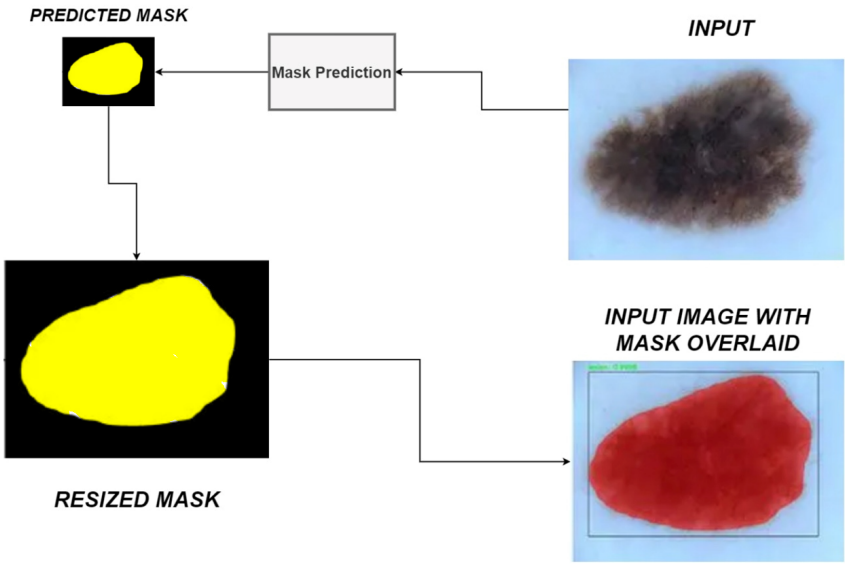

Figure 4.6 – Mask Adding

Mask R-CNN distinguishes from many other models, by isolates items depending on the actual picture and classifies them based on the settings. Many other settings of the item are adjusted throughout classification operations. For instance, interpolation of an object's position, frames of reference, graphic picture of an item, and etc.

Convolutional Neural Networks

Convolutional Neural Networks are extremely similar to the simple neural Networks in that they would be composed of neurons with trainable weights and biases. Every neuron gets input data, is doing a dot product, and then potentially follows it with a non-linearity. From unprocessed picture pixels on one side to class values on the other, the entire network still represents a single differentiable scoring function. They still contain a loss function such as SVM and Softmax on the last one layer, and all of the tips we discovered for learning ordinary Neural Networks applicable.

ConvNet designs explicitly assume that the inputs are pictures, allowing us to embed certain attributes into the framework. It thus makes the forwarding function more convenient to construct and cut down the number of parameters in the network significantly. Convolutional Neural Networks use the notion that the input consists of pictures to limit the design in a more reasonable manner. Besides a traditional Neural Network, ConvNet's layers have neurons organized in three dimensions: width, depth, and height. It should be noted that the term "depth"relates to the third dimension of an activation volume. For instance, the input pictures are an activation intake volume with size 32x32x3 or height, width, depth correspondingly. Furthermore, for the corresponding output layer might have dimensions 1x1x10, since at the completion of the ConvNet architecture, we

would have compressed the whole picture to a vector representation of class scores grouped across the depth dimension. Here's an illustration:

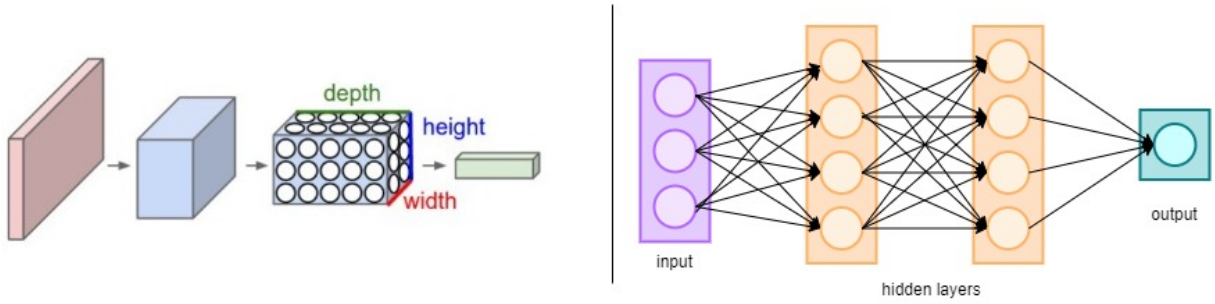

Figure 4.7 – Left:A standard 3-layer Neural Network. Right: A ConvNet organizes its neurons in three dimensions.

ConvNet designs are built with three types of layers: Layer, Pooling Layer, and Fully-Connected Layer. Each Layer takes an input 3D dimension and employs a discrete function to convert it to an output 3D space. Additional hyperparameters may or may not be present in each Layer. ConvNets accomplishes this by layering the original image from the initial pixel values to the resulting class scores.

The UNeT network is a group of people who work together to solve problems. Medical photos were the inspiration for the UNet convolutional network. With a tiny data set, achieves great accuracy and application capabilities. The network is trained end-to-end on a limited number of pictures and outperforms the current best way (convolutional network with sliding window) for segmenting neural structures in electron microscopic stacks in the ISBI competition. UNet won the 2015 ISBI Cell Tracking competition in these categories by a large margin, using the same network that was trained on transmission light microscopy pictures.

The coder-decoder architecture is represented by Unet. By merging layers, the encoder gradually decreases the spatial dimension, while the decoder gradually recovers the object's features and spatial dimension. There are additional quick links between the encoder and the decoder, which aid the decoder in recovering the object's features:

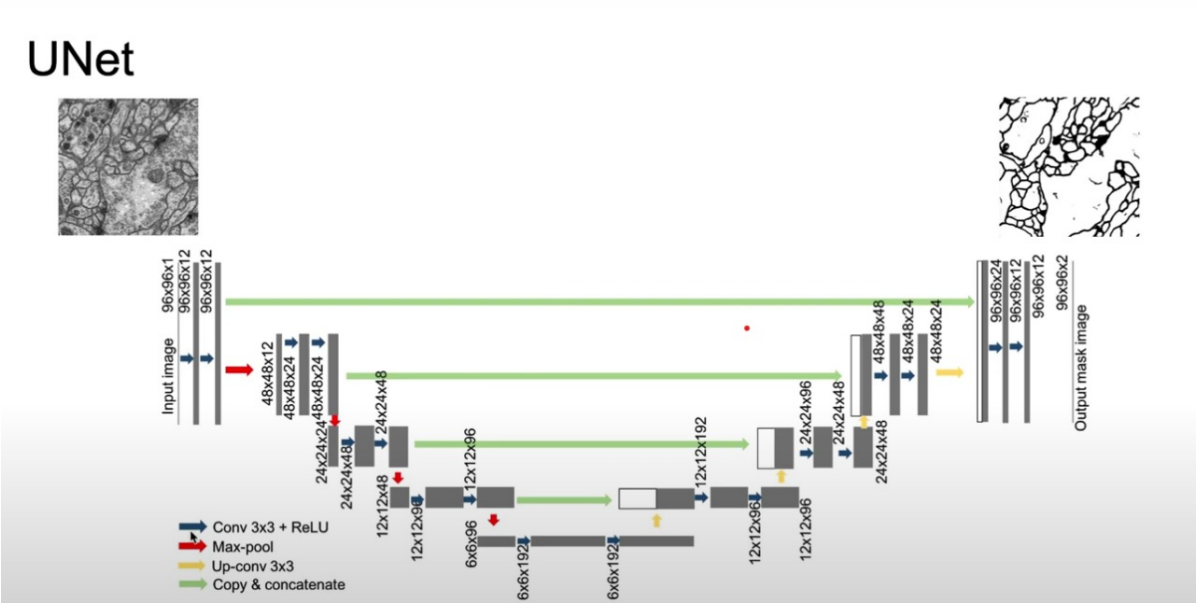

Figure 4.8 – Unet structure

Only its actual layers are used in the network; there have been no fully connected layers. The whole context of each convolution, namely the segmentation map, is presented on the input screen. Increase the amount of data for Unet quality segmentation by distorting existing photos.

5 Software technologies and libraries

Python is a high-level programming language that may be used to create a variety of applications. Python is a programming language that may be used to create web applications, games, and databases. Python is a widely-used programming language in the fields of machine learning and artificial intelligence. Python is a programming language that is interpreted. This means that the source code will be partially translated into machine code when it is read by a particular interpreter. Python has a distinct syntax that distinguishes it from other programming languages. Because Python rarely utilizes auxiliary syntactic features like square brackets and semicolons, reading the code is easier than in other programming languages. To display nested structures, programmers must indent, according to language constraints. A well-written text with few distractions is obviously simple to read and comprehend.

Python is a full-featured general-purpose programming language with applications in a variety of industries. Object-oriented programming is the core paradigm it supports. However, we will just discuss objects and structural programming in this course. Because this is the starting point. There's no use in learning if you don't understand the fundamental data types, branches, loops, and functions because all of the complicated paradigms are utilized. The Python interpreter is available for free under GNU General Public License and other similar licenses.

In comparison to other languages, Python regularly uses pre-built libraries. Tensorflow, Detectron2, and Touchvision are examples of libraries that work with ready-made models. Coco (Common Objects in Context) datasets, RCNN (Region-Based Convolutional Neural Networks) models, and so on are the most common examples. There are numerous COCO models available for now.

AI is made up of two primary learning and immersive modules that interact with Python images, such as matplotlib, numpy, sklearn, keras, and others. Data processing, data analysis, modeling, visualization, and drawing are the main areas of machine learning research in Python. Learning models, data processing, analysis (PANDAS), and data drawing are always used in the development of the library, which is written in Python. If you try to learn Python without knowing the language, it's like trying to learn English without knowing the words, thus Python does it for you.

Python is considerably faster to learn than skimming data with other languages. Python is an industry leader in data analysis and modeling, data processing programs, and data scanning. Scanners: Queries, Scrapy, Selenium, and BeautifulSoup are all libraries that are required for the building of web scanners and will be available in the Python database.

Numpy, Scipy, Pandas, and Matplotlib are data processing libraries that can

be used for matrix calculations, scientific calculations, and drawing, among other things. The simulation technique works in the data stream after the data is in the format. Modeling: nltk, Keras, and sklearn are deep learning, and machine learning libraries. If your project is built on an existing online system, you can learn how to create it in Python for free and make your model available in your system. Python is without a doubt the most popular programming language. I enjoy its simplicity, adaptability, and extension because it is widely used in all fields.

To begin with, Python is a free and open source programming language. This means that the developer can make changes to it if he or she sees fit. The syntax of this programming language is always developing, making it simpler to use and more efficient.

Second, numerous fully prepared libraries can help you write code faster. TensorFlow, for example, is widely used for machine learning and datasets. Scikit is a programming language that can be used to teach machine learning models. PyTorch is a programming language for voice synthesis and computer vision. This is a significant benefit that can help you save time and money by allowing you to employ pre-made solutions rather than generating them from scratch.

Finally, Python is not only cross-platform, but it also plays well with other AI programming languages. Python is a good choice for building little scripts as well as supporting major corporate initiatives, regardless of the scale or size of your project. Python comes extremely close to being a universal AI programming language.

Anaconda: Python Development Environment (version 3.7) As a compiler, I used Anaconda. Anaconda made it simple to implement the application because all I had to do was install and provide basic configuration and based, as well as the required libraries. Anaconda comes with a powerful code editor that includes syntax highlighting, automated formatting, configurable indentation, and more. Anaconda allows you to examine the interoperability of different language translator versions as well as employ code templates. Anaconda gives you the ability to easily restore your code and use a graphical assembler. Anaconda allows you to use Python, Django, SSH, as well as the interactive database interface, as well as perform embedded unit tests. Cross-platform development environment Anaconda: It runs on Linux, Windows, and Mac OS X.

Anaconda is an integrated development environment (IDE) and a unified development environment (UDE). An IDE is a collection of software tools used by programmers to develop software. The way you utilize basic tools like a text editor, compiler, and so on is the polar opposite of how you use an IDE for software development. In contrast to separate development programs, this allows developers to make fewer attempts to move between modes. Development

environments, on the other hand, can only expedite the software development process after additional training because IDEs are complicated software. Many individuals are interactive to reduce input impedance and make the most of a single IDE to make switching from one supplier to another easier.

In most cases, the IDE is the single software in which all development takes place. Many functions for creating, changing, building, deploying, and testing software are frequently included. The integrated environment's goal is to merge multiple utilities into a single module that allows you to abstract from handling auxiliary chores, allowing the programmer to concentrate on solving algorithmic problems rather than wasting time performing mundane technical duties (for example, calling a compiler). As a result, the developer's productivity rises. Strong integration of development tasks is also thought to boost productivity by allowing for the introduction of extra functionality at intermediate stages of work. You can utilize the IDE to examine your code, provide instant feedback, and report syntax mistakes, for example.

The majority of today's IDEs are graphical. However, before they became widely utilized, the first operating systems featuring a graphical ISR interface were employed. These operating systems use a text-based interface that calls multiple functions via functions and hotkeys.

The following programs are included in the development environment:

- text editor,
- Translator
- assembly automation tools,
- Repairman.

Anaconda is a Python development environment for professionals. Anaconda aids in the creation of aesthetically pleasing and maintainable code. To regulate code quality and aid testing, the IDE employs compliance checks, smart refactoring, and numerous inspections.

Anaconda's key advantages over AI Record are as follows:

1 The clever code analysis engine enables precise auto-completion, error detection and correction, code navigation convenience, and other important features. Editor who is astute.
2 Anaconda makes code editing easier with features like code completion, quick code review, error detection and correction, automatic refactoring, and intuitive navigation.
3 Assistance with scientific computing It is possible to run commands on the interactive Python console, attach Anaconda libraries, and function melancholic with other libraries in the interest of scientific computing to offer data analysis, such as Matplotlib to expose NumPy, using Anaconda.
4 Remote development possibilities. With the support of remote interpreters and

built-in SSH terminal interfaces, Anaconda can run, debug, examine, expose, and deploy on remote hosts or virtual machines.

5 Tools for the advantage of programmers. Anaconda integrates a built-in debugger to reveal the tester, as well as a Python profile, a built-in terminal, and database management tools to expose common version control systems.

For the dataset, this program was used, JSON consists of two different structures: like certain data that is stored as a value that has its key, where the keys are given as strings and are not often repeated, everything happens in a standard way. The second structure is a regulated value base, which exists in many programming languages and is described as a vector, list, etc.

In JSON, the requirements should be used as values:

- The entry is an unordered set of key pairs: the value is contained in parenthesis . A key string containing the ":"character is used to describe it. Commas are used to separate key-value pairs. One-dimensional array is a collection of arranged items. Square brackets surround the "[]"array. Commas are used to split values. This doesn't matter if the array is incomplete. Different values can be found in the same array:

- Quantity data

- Boolean data: True, false and null values.

- A string is a comma-separated collection of zero or more Unicode characters surrounded by double-quotes.

- Characters can be provided using a slash "/"escape or in Unicode hexadecimal encoding.

Detectron2 is the second version of the library called Detectron. In the first version, there was only the Mask R-CNN model indicator. At this time, the library is distinguished by its variable design and is implemented in Pytorch. In addition, it can train and test models on one or more GPU servers. Models inside Detectron are separated into modules that can allow changes in model construction without any difficulty. In research papers, particularly can notice similarities in terms of the coding desk. Many of the models from the original library are also included in Detectron2, such as Mask R-CNN, RetinaNet, and DensePose. For object identification, the package allows you to use synchronous batch retrieval and updated datasets. In this case, the library is used for the purpose of recognition of these tasks and also has several purposes for use, such as all types of object segmentation, recognition by boxes, and prediction of human action.

In our case, segmentation methods have been used that work through setting security parameters in this Detectron2 library. Since, of course, this particular method is more relevant and common among the rest. The library can train on the GPU server since the original model has been translated into this server, which

creates speed and efficiency in performing training. Detectron2 is a library that allows you to correctly rewrite the model on a real product. And also, the library has functions for more efficient and uncomplicated functions for standard models and neural networks.

Torchvision is a project part of the Pytorch library. Pytorch is a platform system to work in the field of machine learning. As a whole, the Torchvision package consists of familiar structures of models, working with images for computer vision. PyTorch is created in the Python language and is based on Torch, which provides open source code. Basically, this package is used in work with CV or with NLP. PyTorch produces grouping actions and separation in automatic form. For the calculation of gradients, the record of calculations occurs first, then the reverse occurs. This specific method is considered to be one of the most demanding when working with models, especially when working with neural networks, so it calculates the difference with differential correction. Torchvision is a library that runs in parallel with PyTorch and is the library of computer vision. Here is the primary data, some designs of the image, and the images themselves.

Because each package includes internal packets, this packet also includes vision datasets, which involves transferring new data from existing data. The second package, called vision models, has features in image data manipulation, semantic segmentation, video detection, and object recognition. The last package, torchvision transforms, is the most accessible and easy to use in image changes.

6 Experiment

Let us determine the libraries versions used, while working with the Detectron2 library, the latest version of Python is used and a wide variety of models are built, and some components can be used in the old version, for example, such as:

- OpenCV has a common open-source code that is used in the analysis of any image.

- Torchvision is a parallel PyTorch package for computer vision.

- Matplotlib is a library for visualizing statistical or interactive animation data. It is used for visualization using two-dimensional or 2D graphs; sometimes it is used in the case of 3D.

```
import os
import numpy as np
import json
import random
import matplotlib.pyplot as plt
%matplotlib inline

from detectron2.structures import BoxMode
from detectron2.data import DatasetCatalog, MetadataCatalog
```

Figure 6.1 – Program Libraries

Visualizer, Colormode, get cfg, DefaultTrainer, DefaultPredictor, MetadataCatalog, and model zoo are mostly tools which can only be setup in Detectron2.

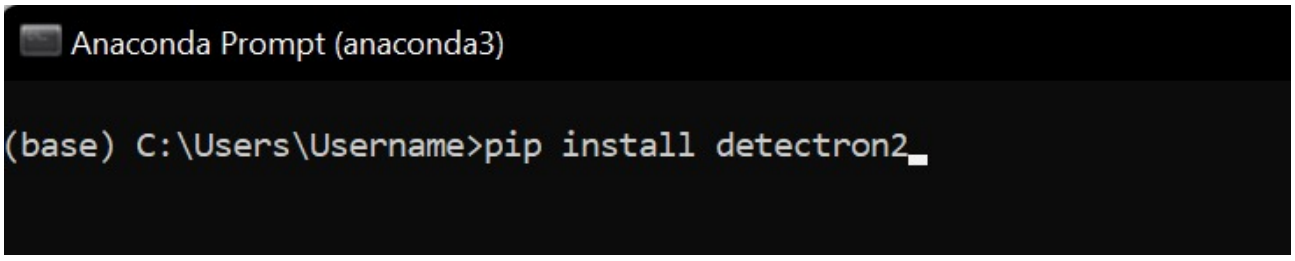

Figure 6.2 – Detectron2 download

The installation of Detectron2 takes place in cmd in the Anaconda application, and further tools used were taken from the internet.

By this command we installed torch1.8 version of torchvision. Because Detectron can work in this variant. Pyyaml module is a YAML 5.1 version, which

is a syntax analyzer of the markup language. Then, we install also 1.8.0 . Here, "0.9.0+cu101 -f " command, where GPU type deployed.

```
import os
import numpy as np
import json
import random
import matplotlib.pyplot as plt
%matplotlib inline

from detectron2.structures import BoxMode
from detectron2.data import DatasetCatalog, MetadataCatalog
```

Figure 6.3 – Program libraries

After that we have imported Detectron2 libraries needed for model,dataset, training, predictors , catalogs, configurations, colors and visualizers. Model-zoo -ready-made model, which works with Mask RCNN network. After pre-loading in the terminal, working with configuration files.

Adding photos to the database

At the beginning, we set the desired photo in the program directory:

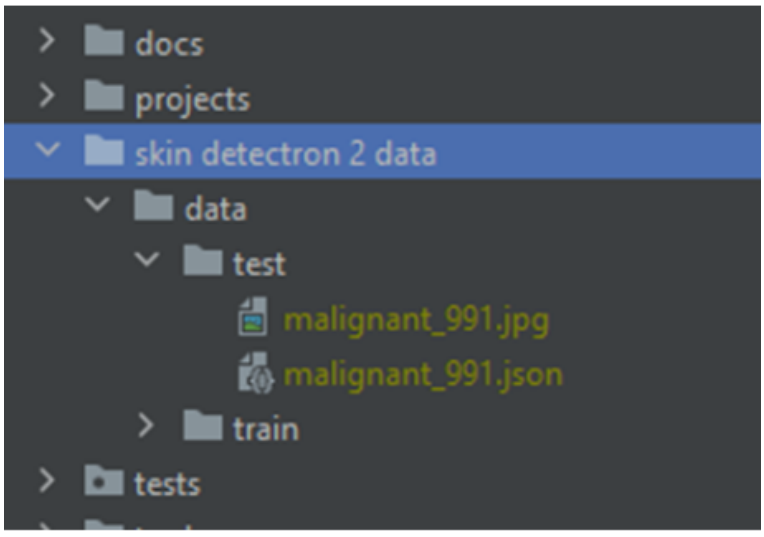

Figure 6.4 – Directory in the database

The preceding is how the process of recognizing images of different types of cancer was done. We have a large number of photos of various types of cancer in the data, which are separated into two categories: "benign" and "malignant". Here, benign refers to a condition that is "not cancer", while malignant refers to

a condition that is “cancer cells”.

Distribution of instances among all 2 categories:

category	#instances	category	#instances
benign	835	malignant	165
total	1000		

Figure 6.5 – Number of Classes

From above table, can be drawn such conclusions, “Benign” - non-malignant-835 photos, “Malignant”-165 photos, in total number of photos-1000.

7 Program explanation

The program code is divided into several branches . The “def run” function in the source tree is a function for setting up configuration settings . The program code is the main model of the Mask – RCNN format . The size of the settings in it does not exceed 231 megabytes.

The size of the JSON file is determined by the implementation of the default parameters. Most parameters, for example, are shown as a collection of standard-type variables. This is undeniably demonstrated by the installation of the detectron2 and torchvision libraries. Parameters, keeping in mind model segmentation, interact with the OBP printer. And file sizes vary according to database execution.

The segmentation system is compatible with the most recent model cascade version. This is absolutely great. Admittedly, the fundamental precept of various models, relying on the diversity of model guidelines, operates at random with configuration settings. Mirrors do tasks connected to the interpretation of the primary focuses, according to the requirements of the simulation system in the library.

```
def get_data_dicts(directory, classes):
    dataset_dicts = []
    for filename in [file for file in os.listdir(directory) if file.endswith('.json')]:
        json_file = os.path.join(directory, filename)
        with open(json_file) as f:
            img_anns = json.load(f)

        record = {}

        filename = os.path.join(directory, img_anns["imagePath"])

        record["file_name"] = filename
        record["height"] = 224
        record["width"] = 224

        annos = img_anns["shapes"]
        objs = []
        for anno in annos:
            px = [a[0] for a in anno['points']] # x coord
            py = [a[1] for a in anno['points']] # y-coord
            poly = [(x, y) for x, y in zip(px, py)] # poly for segmentation
            poly = [p for x in poly for p in x]

            obj = {
                "bbox": [np.min(px), np.min(py), np.max(px), np.max(py)],
                "bbox_mode": BoxMode.XYXY_ABS,
                "segmentation": [poly],
                "category_id": classes.index(anno['label']),
                "iscrowd": 0
            }
            objs.append(obj)
        record["annotations"] = objs
        dataset_dicts.append(record)
    return dataset_dicts
```

Figure 7.1 – Function of dictionary

The main release method develops a framework for identifying objects based on their geometric attributes. This technique is used to register the labelme dataset in general. More specifically, we begin by calling the def get-data-dicts() function from the def-start() function. Where path corresponds to a data path and categories refer to file categories (benign, malignant).

We give a way to the database directory here. We install json files in the filenames that were initially written using the 'For loop'. If the files are present in the directory, if it contains the json, the newly opened json-file will insert references to the variable. Whereas the 'os.path.join' method adds strings while taking into consideration the operating system's characteristics. The with open function is used to read the file system. Next, we open the record dictionary. In this , we enter the file name, length, and width. Files in the database must be 224 x 224 .

In the 'anno' variable, we assign the shapes parameter . Then we open the 'obj array', which is a setting for the photo configuration parameter in the labelme database. In the 'anno' loop, we set the coordinates px and py . Where" px "is the X coordinate, and" py "is the y coordinate . Then we write the poly variable, a polygamous segmentation parameter .

We highlight the essential parameters in the Obj dictionary. Using the detectron2 connection, we established the appropriate segmentation parameters in the first set of collections. This is the Obj dictionary, which is a collection of settings that we require, every file's 'category-id' is recorded. When the Labelme dataset parameters are complete, we append them to our 'dataset-dicts' array.

```
classes = ['benign','malignant']

data_path = '/content/drive/MyDrive/datasets/skin detectron 2 data/data/'

for d2 in ["train", "test"]:
    DatasetCatalog.register(
        "category_" + d2,
        lambda d2=d2: get_data_dicts(data_path+d2, classes)
    )
    MetadataCatalog.get("category_" + d2).set(thing_classes=classes)

microcontroller_metadata = MetadataCatalog.get("category_train")
```

Figure 7.2 – Class distribution

In the following line, we provide two input class labels. Then, in the attribute data path, we create a connection to the contents' location. The above approach generates data. This cycle is usually installed in the methadone. To implement the connect section, write get data dicts here. As Metadays reveals details on the

qualities and properties that characterize each thing, it can begin to explore and manage massive amounts of data efficiently. Developing a metadata store is the basis for the parameter microcontroller metadata. The number of images in the categories "cycling,general test,"and "train"is displayed.

```
cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("category_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml")
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 1000
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 2
```

Figure 7.3 – Program config

The CFG setup parameters are set there:

cf.merge-from-file - COCO for downloading data by a directory (temporal segmentation of modeling techniques);

cfg.dataloader.num-workers = Rate at which content is downloaded into a buffer for the first time;

cfg.model.weights=model zoo.get-checkpoint-url A management criterion that could save the model and other milestone elements; cfg.datasets.train = category train settings reference; cfg.model.roi-heads.num-classes = amount of images inside the category

Neural network-based training

This type of methodology is used to train the neural network within the software. The training technique in the Detectron2 package is the basic SNP algorithm. We utilize only a bit of code to retrieve personal details in COCO JSON files. which is the only script direction that will allow us to launch projects.

```
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
```

Figure 7.4 – Training code for neural networks

After learning the NN model, the runtime , iteration time , and total amount of time are displayed . In the configuration settings, we set the Predicat configuration to DefaultPredictor(cfg),this is a standard configuration of the Detectron2 library . test-dataset-dicts is written in dictionary mode for the dataset we downloaded . Into this we write the folder route of our dataset.

```
trainer.train()
```

```
[05/30 09:34:16 d2.engine.train_loop]: Starting training from iteration 0
[05/30 09:34:24 d2.utils.events]:  eta: 0:06:12  iter: 19  total_loss: 1.889  loss_cls: 1.046  loss_box_reg: 0.1468  loss_mask:
0.6905  loss_rpn_cls: 0.002486  loss_rpn_loc: 0.01103  time: 0.3767  data_time: 0.0170  lr: 4.9953e-06  max_mem: 2754M
[05/30 09:34:31 d2.utils.events]:  eta: 0:05:51  iter: 39  total_loss: 1.677  loss_cls: 0.8557  loss_box_reg: 0.1151  loss_mas
k: 0.687  loss_rpn_cls: 0.007567  loss_rpn_loc: 0.009329  time: 0.3615  data_time: 0.0058  lr: 9.9902e-06  max_mem: 2754M
[05/30 09:34:39 d2.utils.events]:  eta: 0:05:43  iter: 59  total_loss: 1.3  loss_cls: 0.5175  loss_box_reg: 0.1214  loss_mask:
0.6754  loss_rpn_cls: 0.008588  loss_rpn_loc: 0.009788  time: 0.3646  data_time: 0.0072  lr: 1.4985e-05  max_mem: 2754M
[05/30 09:34:46 d2.utils.events]:  eta: 0:05:38  iter: 79  total_loss: 1.117  loss_cls: 0.3139  loss_box_reg: 0.1288  loss_mas
k: 0.6626  loss_rpn_cls: 0.01042  loss_rpn_loc: 0.008777  time: 0.3672  data_time: 0.0071  lr: 1.998e-05  max_mem: 2754M
[05/30 09:34:54 d2.utils.events]:  eta: 0:05:31  iter: 99  total_loss: 1.023  loss_cls: 0.206  loss_box_reg: 0.1273  loss_mask:
0.6414  loss_rpn_cls: 0.005921  loss_rpn_loc: 0.012  time: 0.3686  data_time: 0.0088  lr: 2.4975e-05  max_mem: 2754M
[05/30 09:35:01 d2.utils.events]:  eta: 0:05:24  iter: 119  total_loss: 0.9547  loss_cls: 0.1732  loss_box_reg: 0.1253  loss_ma
sk: 0.6281  loss_rpn_cls: 0.005638  loss_rpn_loc: 0.009367  time: 0.3693  data_time: 0.0077  lr: 2.997e-05  max_mem: 2754M
[05/30 09:35:08 d2.utils.events]:  eta: 0:05:17  iter: 139  total_loss: 0.9066  loss_cls: 0.1569  loss_box_reg: 0.1385  loss_ma
sk: 0.6069  loss_rpn_cls: 0.007453  loss_rpn_loc: 0.009366  time: 0.3695  data_time: 0.0063  lr: 3.4965e-05  max_mem: 2754M
[05/30 09:35:16 d2.utils.events]:  eta: 0:05:11  iter: 159  total_loss: 0.9027  loss_cls: 0.154  loss_box_reg: 0.133  loss_mas
k: 0.5774  loss_rpn_cls: 0.005017  loss_rpn_loc: 0.008669  time: 0.3714  data_time: 0.0066  lr: 3.996e-05  max_mem: 2754M
[05/30 09:35:24 d2.utils.events]:  eta: 0:05:04  iter: 179  total_loss: 0.876  loss_cls: 0.1605  loss_box_reg: 0.1567  loss_mas
k: 0.5199  loss_rpn_cls: 0.001812  loss_rpn_loc: 0.009322  time: 0.3729  data_time: 0.0077  lr: 4.4955e-05  max_mem: 2754M
[05/30 09:35:31 d2.utils.events]:  eta: 0:04:58  iter: 199  total_loss: 0.8428  loss_cls: 0.1305  loss_box_reg: 0.1392  loss_ma
sk: 0.5222  loss_rpn_cls: 0.002791  loss_rpn_loc: 0.006508  time: 0.3734  data_time: 0.0065  lr: 4.995e-05  max_mem: 2754M
[05/30 09:35:39 d2.utils.events]:  eta: 0:04:51  iter: 219  total_loss: 0.8026  loss_cls: 0.1331  loss_box_reg: 0.1432  loss_ma
sk: 0.4737  loss_rpn_cls: 0.002963  loss_rpn_loc: 0.007121  time: 0.3738  data_time: 0.0071  lr: 5.4945e-05  max_mem: 2754M
[05/30 09:35:47 d2.utils.events]:  eta: 0:04:44  iter: 239  total_loss: 0.7319  loss_cls: 0.1367  loss_box_reg: 0.1658  loss_ma
sk: 0.4366  loss_rpn_cls: 0.00156  loss_rpn_loc: 0.0075  time: 0.3760  data_time: 0.0076  lr: 5.994e-05  max_mem: 2754M
[05/30 09:35:55 d2.utils.events]:  eta: 0:04:38  iter: 259  total_loss: 0.67  loss_cls: 0.1088  loss_box_reg: 0.125  loss_mask:
0.398  loss_rpn_cls: 0.002091  loss_rpn_loc: 0.004835  time: 0.3769  data_time: 0.0071  lr: 6.4935e-05  max_mem: 2754M
```

Figure 7.5 – Neural network analysis

```
[05/30 09:33:36 d2.engine.defaults]: Model:
GeneralizedRCNN(
  (backbone): FPN(
    (fpn_lateral2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral3): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral4): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output4): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (fpn_lateral5): Conv2d(2048, 256, kernel_size=(1, 1), stride=(1, 1))
    (fpn_output5): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (top_block): LastLevelMaxPool()
    (bottom_up): ResNet(
      (stem): BasicStem(
        (conv1): Conv2d(
          3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False
          (norm): FrozenBatchNorm2d(num_features=64, eps=1e-05)
        )
      )
```

Figure 7.6 – Text-based learning setting of the Mask-RCNN model before training

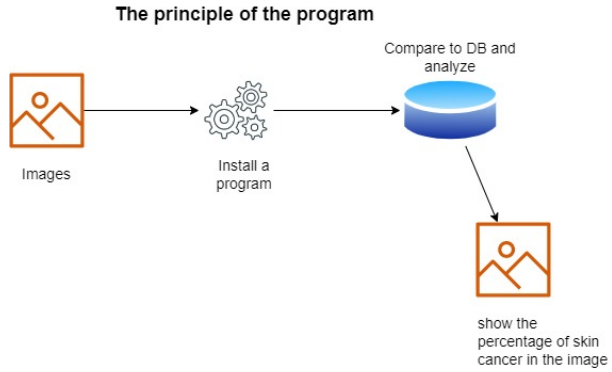

Figure 7.7 – The visual program's operation concept.

From the above chart, it can be viewed the basic principle of the project.

8 Results

After the program has been fully trained, we repeat the For loop. The called Loop interacts with the AI's microcontroller, the zoo model in the Detectron2 library, and the resource configuration management files in the metadata settings.

The IMG variable is used for predicate sharpening. The microcontroller setup is directly in the segmentation-oriented simulation system in the Detectron2 library. We achieve the appropriate photo settings by changing the image size in the Visulizer section. The visualization portion focuses on predicate classification, identifying the torchvision bookshelves as the original model.

In the dataset, category test, this loop obtains photographs at random. It runs the Our generated data may be utilized to segmentation algorithms to perform equally on the difficult medical dataset. We also exhibited an application that included segmentation and percentage of skin cancer estimation. We suggest that in many cases, the time-consuming and labor-intensive job of capturing and annotating real datasets may be decreased or avoided entirely in favor of synthetically created scenarios.

As proven by our generalization tests, one limitation is the reliance on high-quality image input. While existing gaps between the training set and real objects can be filled by new real-world photos from video, a wholly synthetic approach would be preferred. Furthermore, our arranging approach chooses random images, whereas often organized items in practical ways. More sophisticated arrangement possibilities might be developed to represent this tendency.received photographs through the OpenCV library, which he uses to create an urgent predicate. The visualization method is displayed using the Visualizer variable. The risk of benign, if not deadly, tumors is plainly obvious when the ensuing az result is output by an urgent parameter.

Here is the ColorMode.Imagebw displays photos from the black and white side. Using Instance-predictions, we transfer the necessary emphasis to the GPU . With the GPU, the program execution speed increases significantly . With Plt, we can see the desired limited objects in the photo with a size of 14 x 10. Using Color-BGR2RGB, objects are randomly filled with an RGB color palette. With the run() function we can setup our program.

Furthermore, by evaluation of Mean average precision, it achieves the top result on our generated dataset with a mAP of 67.8 %.

Finally, there can be seen screen of results with a percentage degree whether the tumors are benign or not malignant:

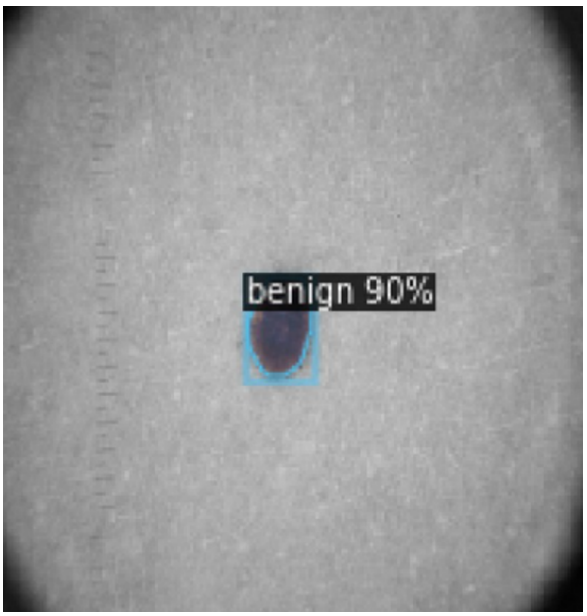

Figure 8.1 – Results

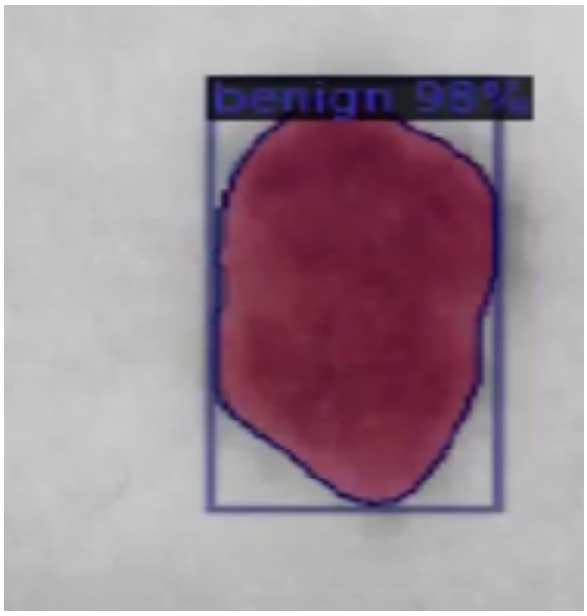

Figure 8.2 – Results

9 Discussion

Our generated data may be utilized to segmentation algorithms to perform equally on the difficult medical dataset. We also exhibited an application that included segmentation and percentage of skin cancer estimation. We suggest that in many cases, the time-consuming and labor-intensive job of capturing and annotating real datasets may be decreased or avoided entirely in favor of synthetically created scenarios.

As proven by our generalization tests, one limitation is the reliance on high-quality image input. While existing gaps between the training set and real objects can be filled by new real-world photos from video, a wholly synthetic approach would be preferred. Furthermore, our arranging approach chooses random images, whereas often organized items in practical ways. More sophisticated arrangement possibilities might be developed to represent this tendency.

10 Conclusion

This project's core concept was as follows. In a challenging period for medicine, with a growth in new illnesses and sick people, it is extremely difficult to find a high-tech solution that works swiftly, with no errors, is associated with high accuracy, and, most importantly, detection accuracy. We can generate products of high quality by using artificial intelligence technology and the establishment of an artificial neural network. Artificial intelligence technologies can help patients with weakened immune systems save money and identify illness more quickly.

Artificial intelligence is becoming a vital aspect of medicine in this era of rapid technological advancement. Artificial intelligence, according to many scientists throughout the world, has considerably increased the accuracy of disease diagnosis products. It's fantastic. People who suffer from mental illnesses are more likely to experience depression, anxiety, and fear of the future. You can avoid such agony by consulting a doctor ahead of time. In most cases, susceptibility to the disease can be identified when it comes to skin tumors. They are generally non-harmful and non-dangerous. However, there are some disorders worth paying attention to. Doctors and patients benefit from artificial intelligence, machine learning, and neural network technologies. Medical advancements will help us to precisely identify ailments, find treatments promptly, and keep track of a patient's status. These are just a few of the benefits that artificial intelligence has brought to the healthcare business.

AI in medicine the integration of algorithms and software to bring human understanding closer to the interpretation of complicated medical data. MRI imaging, ultrasound findings, cardiograms, computed tomography, and cancer diagnosis all employ AI technology. With our application, you may predict the onset of serious diseases like cancer and discover a cause to consult oncologists. This provides promise for the early diagnosis of tumor illness and the treatment of the disease at its origin.

Let us discuss note the results obtained during the work:

1 We looked into the process of creating and training artificial neural networks, as well as the architecture of neural networks, and came up with a viable model. According to the command Librarydetectron2, the neural network's major goal in this project was self-learning. The premise behind neural networks is that they read the configuration and exchange it with controls that ensure that programming runs smoothly. The MaskRCNN configuration has been successfully adopted by the model. The model was given the relevant configuration parameters by the architecture.

2 Developed an artificial neural network based on processing of configuration parameters, including the implementation of COCO model. Due to that, the neural network was able to read databases created using the json model in

about 15-20 minutes.

3 The database contains a significant number of photographs of malignant neoplasms of various types. More than 100 photographs have been submitted to our database. The classification of items among them was split into malignant and non-malignant. Tumors have a range of features, ranging from less harmful to more deadly. The photographs were captured with a modest 224 × 224 camera. A neural network evaluated the database in roughly 20 minutes.

4 Cancer photographs of various varieties are categorized. The mask - RCNN model segmentation technique enabled the categorization of malignant tumors. In particular, the setup is controlled by the microcontroller via meta-data. The detectron2 library was used to collect configuration parameters. The percentage metric indicated the accuracy of estimating the likelihood of malignancy.

5 Systems for classifying things based on geometric characteristics have been developed; for example, it has been determined if skin neoplasms are benign or non-cancerous. High accuracy in detecting the likelihood of cancer has been obtained because of the Torch Vision library. The flare vision library's parameters were in charge of classifying things based on geometric aspects and defining common disorders.

For the second model, UNET neural network has been used in the past. Solves the challenge of medical image segmentation. The feasibility of employing this technique for semantic picture processing has been demonstrated by experimental results. Other findings suggest that the UNET network learns faster and collaborates effectively with the Adam optimizer, but that modifying the loss function has minimal impact on the outcomes.

11 Future Work

In the future, we will concentrate on several elements that will improve the accuracy of prediction estimates. Furthermore, Mask R-CNN should be able to be sped up in all kinds of scenarios. We expect that the quick train and test speeds, as well as the framework's adaptability and accuracy, will improve and simplify future instance segmentation research. In future study, we intend to improve the presented technique by combining several methods of characterizing key video objects, such as movements, semantic tags, and sensory processing.

Furthermore, by including the recursive process, we will attempt to construct a fully end-to-end dynamical model for main video object segmentation. Finally, various potential directions and challenges are presented as guidance for future work in object identification and corresponding neural network-based learning methods.

Our future work is maintained on the mission of improving object segmentation in the video stream in the digital development of Kazakhstan by 23rd June 2023.

12 Acknowledgment

We thank the medical center “I-Clinic” for providing us with a real-time video of skin.

BIBLIOGRAPHY

1 Fast online object tracking and segmentation: A unifying approach / Qiang Wang, Li Zhang, Luca Bertinetto et al. // Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. — 2019. — Pp. 1328–1338.

2 Primary video object segmentation via complementary CNNs and neighborhood reversible flow / Jia Li, Anlin Zheng, Xiaowu Chen, Bin Zhou // Proceedings of the IEEE international conference on computer vision. — 2017. — Pp. 1417–1425.

3 *Wenguan Wang Jianbing Shen, Ling Shao.* Video Salient Object Detection via Fully Convolutional Networks / Ling Shao Wenguan Wang, Jianbing Shen // *IEEE Transactions on Image Processing.* — 2018. — Pp. 38–49.

4 Vision-based vehicle detecting and counting for traffic flow analysis / Zhimei Zhang, Kun Liu, Feng Gao et al. // 2016 International Joint Conference on Neural Networks (IJCNN) / IEEE. — 2016. — Pp. 2267–2273.

5 Rvos: End-to-end recurrent network for video object segmentation / Carles Ventura, Miriam Bellver, Andreu Girbau et al. // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. — 2019. — Pp. 5277–5286.

6 *Adak, Saptakatha.* VidSeg-GAN: Generative Adversarial Network for Video Object Segmentation Tasks / Saptakatha Adak, Sukhendu Das // Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing. — 2018. — Pp. 1–9.

7 You only look once: Unified, real-time object detection / Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — Pp. 779–788.

8 Medical Image Segmentation Using Deep Learning: A Survey. / Tao Lei, Risheng Wang, Yong Wan et al. — 2020.

9 *Mortazi, Aliasghar.* Optimization Algorithms for Deep Learning Based Medical Image Segmentations / Aliasghar Mortazi. — 2019.

10 *Zhang, Dong.* Deep Reinforcement Learning in Medical Object Detection and Segmentation / Dong Zhang. — 2020.

11 Augmented reality meets deep learning for car instance segmentation in urban scenes / Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder et al. // British machine vision conference. — Vol. 1. — 2017. — P. 2.

12 Image segmentation using deep learning: A survey / Shervin Minaee, Yuri Y Boykov, Fatih Porikli et al. // *IEEE transactions on pattern analysis and machine intelligence.* — 2021.

13 *Haque, Intisar Rizwan I.* Deep learning approaches to biomedical image

segmentation / Intisar Rizwan I Haque, Jeremiah Neubert // *Informatics in Medicine Unlocked*. — 2020. — Vol. 18. — P. 100297.

14 A review of deep-learning-based medical image segmentation methods / Xiangbin Liu, Liping Song, Shuai Liu, Yudong Zhang // *Sustainability*. — 2021. — Vol. 13, no. 3. — P. 1224.

15 *Peng, Jialin*. Medical image segmentation with limited supervision: A review of deep network models / Jialin Peng, Ye Wang // *IEEE Access*. — 2021.

16 *Козлова, ОВ*. U-net для решения задачи сегментации медицинских изображений / ОВ Козлова, ЕЮ Куница, ММ Лукашевич. — 2019.

17 *Козлова, ОВ*. U-net для решения задачи сегментации медицинских изображений / ОВ Козлова, ЕЮ Куница, ММ Лукашевич. — 2019.

18 *Махфуд, УЛЬД АХМЕД ТАЛЕБ*. Комбинированные алгоритмы сегментации цветных изображений / УЛЬД АХМЕД ТАЛЕБ Махфуд et al. — 2002.

19 *Vladimirovic, Mitrofanov Egor*. Segmentation of objects of interest in a video stream / Mitrofanov Egor Vladimirovic. — 2020.

20 *Попов, ВС*. Бинаризация и сегментация изображений: определение, краткая классификация и осуществление в среде LabVIEW / ВС Попов, Х Дженгиз // *Молодежный научно-технический вестник*. — 2014. — no. 11. — Pp. 44–44.

Appendix A Code listing

Here, full code scripts are shown:

```
!pip install pyyaml==5.1
!pip install torch==1.8.0+cu101 torchvision==0.9.0+cu101 -f https://download.pytorch.org/whl/torch_stable.html
#install old version of pytorch since detectron2 hasn't released packages for pytorch 1.9
```

```
!pip install detectron2 -f https://dl.fbaipublicfiles.com/detectron2/wheels/cu101/torch1.8/index.html
# After this step it will ask you to restart the runtime, please do it.
```

```
import torch
assert torch.__version__.startswith("1.8")
import torchvision
import cv2
```

```
import os
import numpy as np
import json
import random
import matplotlib.pyplot as plt
%matplotlib inline

from detectron2.structures import BoxMode
from detectron2.data import DatasetCatalog, MetadataCatalog
```

```
def get_data_dicts(directory, classes):
    dataset_dicts = []
    for filename in [file for file in os.listdir(directory) if file.endswith('.json')]:
        json_file = os.path.join(directory, filename)
        with open(json_file) as f:
            img_anns = json.load(f)

        record = {}

        filename = os.path.join(directory, img_anns["imagePath"])

        record["file_name"] = filename
        record["height"] = 224
        record["width"] = 224

        annos = img_anns["shapes"]
        objs = []
        for anno in annos:
            px = [a[0] for a in anno['points']] # x coord
            py = [a[1] for a in anno['points']] # y-coord
            poly = [(x, y) for x, y in zip(px, py)] # poly for segmentation
            poly = [p for x in poly for p in x]

            obj = {
                "bbox": [np.min(px), np.min(py), np.max(px), np.max(py)],
                "bbox_mode": BoxMode.XYXY_ABS,
                "segmentation": [poly],
                "category_id": classes.index(anno['label']),
                "iscrowd": 0
            }
            objs.append(obj)
        record["annotations"] = objs
        dataset_dicts.append(record)
    return dataset_dicts
```

Appendix B Code listing

```
classes = ['benign','malignant']

data_path = '/content/drive/MyDrive/datasets/skin detectron 2 data/data/'

for d2 in ["train", "test"]:
    DatasetCatalog.register(
        "category_" + d2,
        lambda d2=d2: get_data_dicts(data_path+d2, classes)
    )
    MetadataCatalog.get("category_" + d2).set(thing_classes=classes)

microcontroller_metadata = MetadataCatalog.get("category_train")
```

```
from detectron2 import model_zoo
from detectron2.engine import DefaultTrainer, DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import ColorMode, Visualizer
```

```
cfg.MODEL.WEIGHTS = os.path.join(cfg.OUTPUT_DIR, "model_final.pth")
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5
cfg.DATASETS.TEST = ("skin_test", )
predictor = DefaultPredictor(cfg)
```

```
test_dataset_dicts = get_data_dicts(data_path+'test', classes)
```

```
for d in random.sample(test_dataset_dicts, 2):
    img = cv2.imread(d["file_name"])
    outputs = predictor(img)
    v = Visualizer(img[:, :, ::-1],
                   metadata=microcontroller_metadata,
                   scale=0.8,
                   instance_mode=ColorMode.IMAGE_BW # removes the colors of unsegmented pixels
    )
    v = v.draw_instance_predictions(outputs["instances"].to("cpu"))
    plt.figure(figsize = (14, 10))
    plt.imshow(cv2.cvtColor(v.get_image()[:, :, ::-1], cv2.COLOR_BGR2RGB))
    plt.show()
```