**L09 Object Detection using Transfer learning and Pascal VOC 207 Dataset**

**Team EyeConic**

**Brooke Broderick - W217065417**

**Matthew Choo - W206744145**

**Cameroun White-W216530571**

**Melvis Maduagwu-W216890463**

**Erick Banegas - W2162009**

**NancyChieu-W000098778**

**Yoana Cook - W215913890**

**6252-ITAI-1378-Comp Vision-Artificial Intel-RT-15698 - Spring 2025**

**Professor Anna Devarakonda**

**March 22, 2025**

**Intro:**

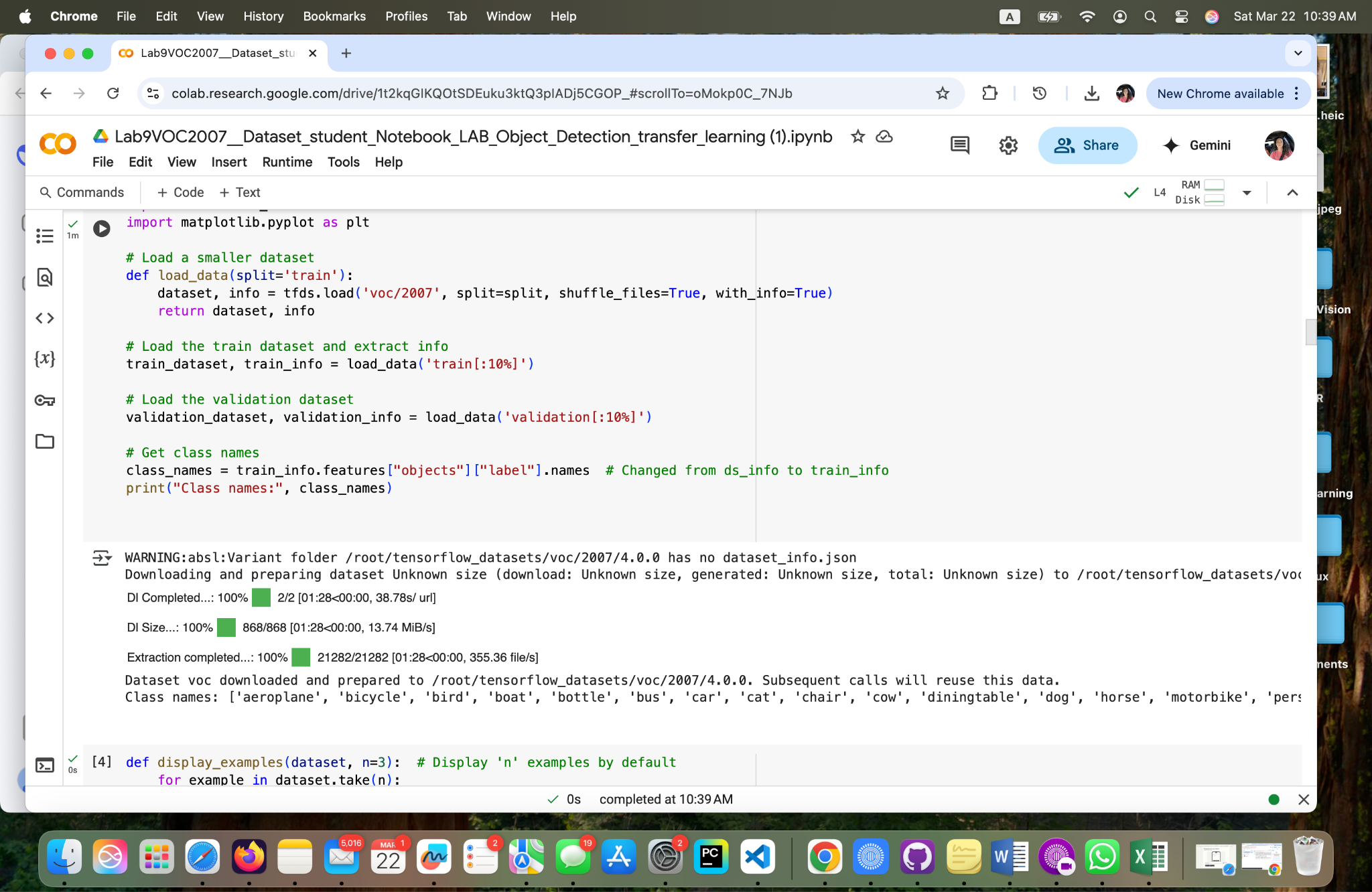
* In this lab, we will be learning the use of object detection by using TensorFlow and Pascal VOC. We will be working on a program called “Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning (1).ipynb”. In this file, we will have to adapt to our image classification task that can be an object detection task. This type of task does not only classify objects within an image but also localizes them with bounding boxes. We will be also using different runtimes which are GPU and TPU. GPU (Graphics Processing Unit) is a specialized processor that is designed to handle the rapid processing and rendering the graphics, images and videos. TPU (Tensor Processing Unit) that specialized application-specific integrated circuits (ASIC) that was created by Google to accelerate machine learning workloads that involves large-scale neural network training and inference. In figures 1 - 11, we have to run the code that is given but have to choose two different runtimes such as T4 GPU or v2-8 TPU.

**Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning (1).ipynb Original Code**

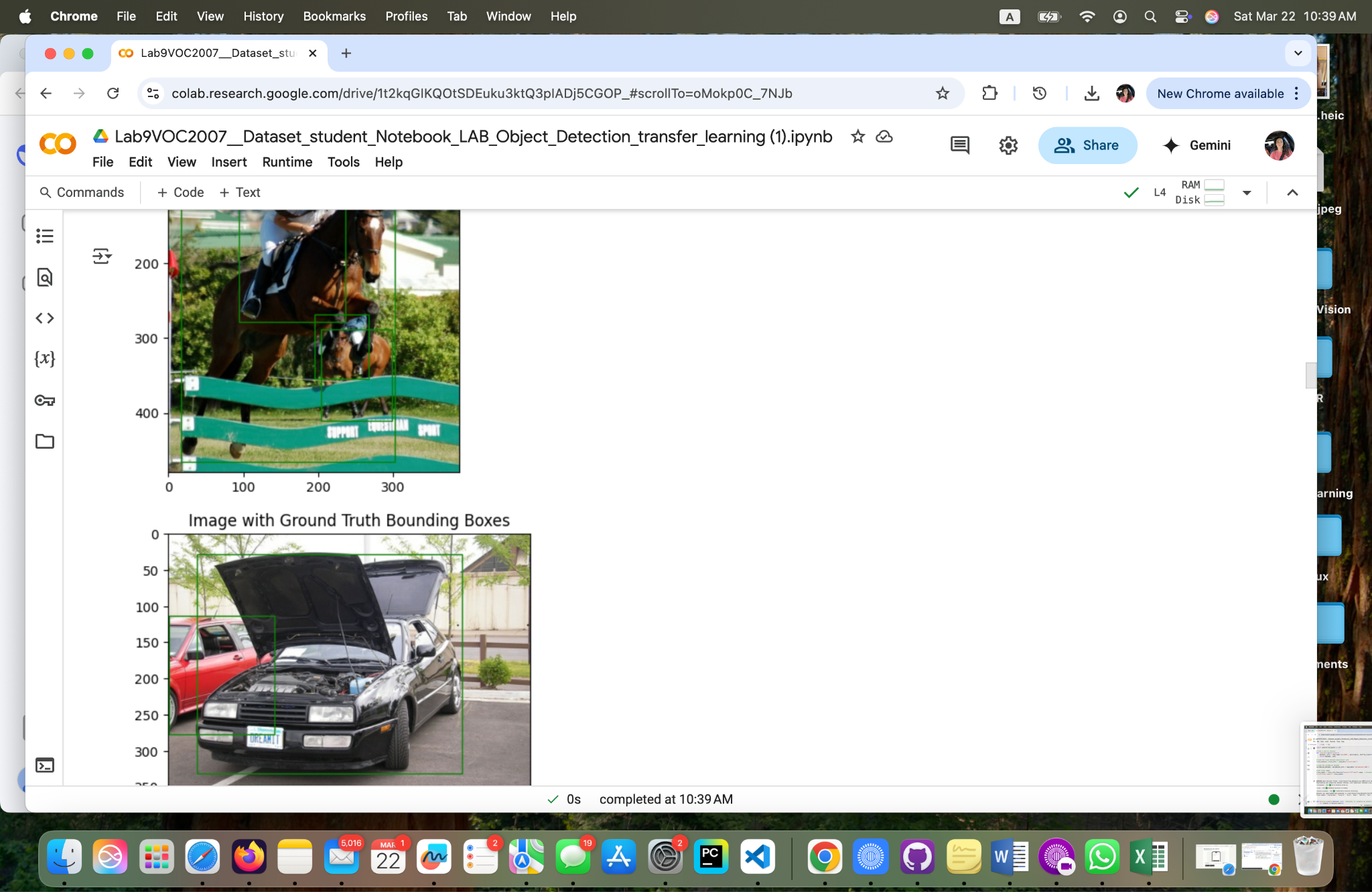
**Yoana Cook**

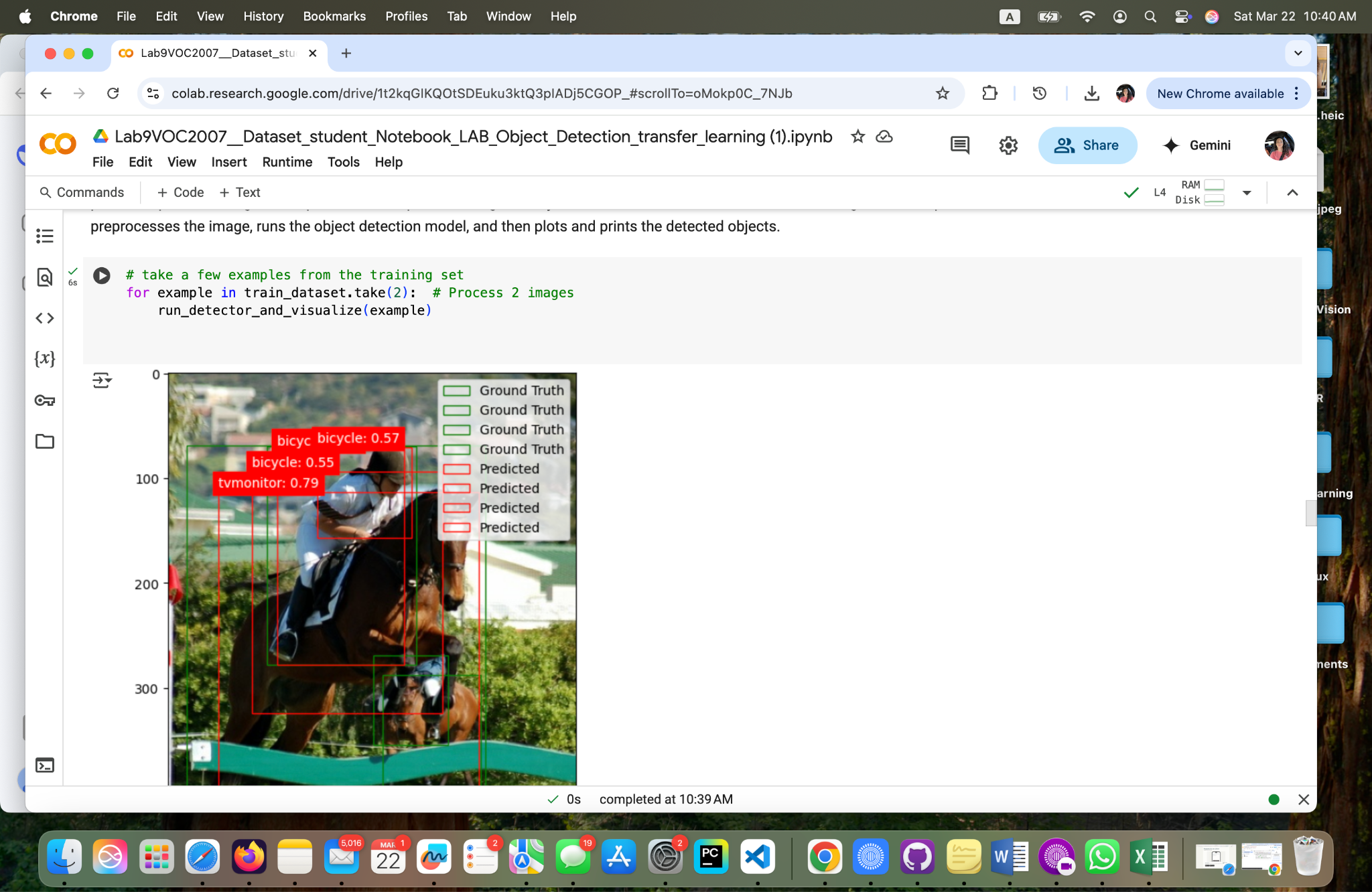
****

**Fig1. Importing and installing the necessary libraries.**

****

**Fig 2. Load the VOC2007 dataset**

****

****

**Fig 4. Processing and Displaying Images with Detections**

**Personal Reflection**

1. **Conceptual Understanding:**

* **What is the main difference between image classification and object detection? How is this difference evident in the output of this exercise?**

**Ans:**

**Difference between Image Classification and Object Detection:**

**Image classification assigns a single label to an entire image.**

**Object detection not only classifies objects within an image but also localizes them using bounding boxes.**

**In the output, this difference is evident as object detection produces multiple labels per image with bounding box coordinates, unlike classification which gives a single label.**

* **Explain why we chose the SSD MobileNet V2 model for this task. What are its advantages and limitations, especially in the context of limited computational resources?**

**Ans:**

**Advantages:**

**Lightweight and optimized for real-time object detection.**

**Efficient on devices with limited computational resources.**

**Pre-trained on COCO, allowing for transfer learning with fewer data points.**

**Limitations:**

**Lower accuracy compared to heavier models like Faster R-CNN.**

**Struggles with small or overlapping objects due to its single-shot detection method.**

1. **Code Interpretation:**

* **Describe the role of the find\_images\_with\_classes function. Why is it useful when working with a large dataset like COCO?**

**Ans:**

**Role of find\_images\_with\_classes function:**

**Helps filter images containing specific object classes.**

**Useful for working with large datasets like COCO to focus on a subset of interest.**

* **In the plot\_detections function, how does the threshold value (threshold=0.5) impact the number of objects displayed?**

**Ans:**

**Effect of threshold=0.5 in plot\_detections:**

**It determines the confidence level required to display a detected object.**

**A higher threshold shows only high-confidence detections, while a lower threshold includes more objects (but with higher chances of false positives).**

* **Explain how the heatmap visualization helps you understand the model's confidence in its detections.**

**Ans:**

**Purpose of Heatmap Visualization:**

**It visually represents the model's confidence in detecting objects at different locations.**

**Helps identify areas where the model is more certain versus regions with uncertain predictions.**

1. **Observing Results and Limitations:**

* **Run the exercise multiple times. Which types of objects does the model tend to detect more accurately? Which ones are more challenging? Can you explain why?**

**Ans:**

**Objects Detected More Accurately vs. Challenging Ones:**

**Likely larger objects (like cars, people) are detected well.**

**Smaller objects or those in cluttered scenes are harder to detect due to SSD’s limitations.**

* **Observe the bounding boxes. Are there any instances where the boxes are inaccurate or miss the object entirely? What factors in the images might be contributing to these errors?**

**Ans:**

**Bounding Box Accuracy Issues:**

**Some boxes may be inaccurate or missing due to:**

**Poor lighting or occlusions.**

**Small object size.**

**Low-quality dataset subset.**

* **How would you expect the accuracy of the model to change if we had used the entire Pascal VOC 2007 dataset instead of a small subset? Why?**

**Ans:**

**Impact of Using Full Pascal VOC 2007 Dataset:**

**Accuracy would likely improve as the model gets exposed to more diverse examples.**

**However, it would require more computational resources.**

1. **Critical Thinking:**

* **How could you modify the code to detect a specific set of objects, like only animals or only vehicles?**

**Ans:**

**Modifying Code for Specific Objects (e.g., only animals or vehicles):**

**Adjust find\_images\_with\_classes to filter images containing specific labels (e.g., 'dog', 'cat' for animals).**

* **If you wanted to train your own object detection model, what steps would you need to take? What are some challenges you might encounter?**

**Ans:**

**Steps to Train a Custom Object Detection Model:**

**Collect and label the dataset.**

**Choose a model architecture (e.g., Faster R-CNN, YOLO).**

**Train using transfer learning or from scratch.**

**Optimize with data augmentation and hyperparameter tuning.**

**Deploy and evaluate performance.**

* **Given the limitations of this model, in what real-world scenarios might it still be useful for object detection?**

**Ans:**

**Real-World Use Cases Despite Model Limitations:**

**Real-time applications (e.g., surveillance, mobile apps).**

**Robotics and autonomous systems with resource constraints.**

**Quick prototyping for object detection tasks.**

1. **Going Further (Optional): (Bonus points)**

* **Research other object detection models available in TensorFlow Hub. Compare and contrast them with SSD MobileNet V2 in terms of accuracy, speed, and resource requirements.**

**Ans:**

**Comparing Other TensorFlow Object Detection Models:**

**Faster R-CNN: More accurate but slower.**

**YOLO: Faster but less precise for small objects.**

* **Try running a few images through a more powerful object detection model online (if available). Compare the results to the output of this exercise. What differences do you notice?**

**Ans:**

**Comparing with a More Powerful Model:**

**Running the same images through a model like EfficientDet or Faster R-CNN would show:**

**More accurate detections.**

**Better handling of small/overlapping objects.**

**Higher computational cost.**

* **Important: Remember, the goal here isn't perfect accuracy. It's to understand the core concepts of object detection, the limitations of working with restricted resources, and how to critically analyze the results.**

**Matthew Choo**

* **After running the code and choosing one of the runtimes such as T4 GPU, I begin to understand the purpose of using these runtimes. T4 GPU is a low power GPU that is designed for data centers that is specifically being optimized for AI inference and virtualized workloads. After running a specific code that was given (Fig 13) was able to see results from the dataset being given. Was able to understand how the images were filtered out. From figures 12 and 32, shows the results of the given code and dataset.**

**Cameroun White**

* **In lab L09, I explored object detection using transfer learning with the SSD MobileNet V2 model and the Pascal VOC 2007 dataset. This was my first hands-on experience applying a pre-trained deep learning model to detect and classify objects in images. Setting up the environment and ensuring GPU support taught me how important hardware acceleration is when training deep learning models. I also learned how TensorFlow Hub allows us to swiftly load powerful models with minimal setup.One of the challenges I faced was understanding how the data pipeline worked,from loading images and annotations to feeding them into the model. However, running the notebook cell by cell and reviewing the visual results helped me better grasp how object detection models make predictions. This lab gave me a better understanding of how deep learning can be adapted to real-world computer vision tasks.**

**Melvis Maduagwu**

1. **Conceptual Understanding:**

**- What is the main difference between image classification and object detection? How is this difference evident in the output of this exercise?**

**Ans**

**In image classification, Models identify what an image is but do not specify if it contains objects or not. In object detection, Models can identify an image and also recognise objects within the image by drawing bounding boxes around them. It's evident in this exercise because image classification has a single label while object detection has different labels, depending on how many objects the model finds in the image. Object detection also has confidence scores.**

**- Explain why we chose the SSD MobileNet V2 model for this task. What are its advantages and limitations, especially in the context of limited computational resources?**

**Ans**

**Its Advantages**

**We chose the SSD MobileNet V2 model for this task because we have limited computational resources and this is a lightweight model, suitable for this task. The SSD (Single Shot MultiBox Detector) MobileNet V2 model was chosen for this object detection task because it provides a decent balance between speed and accuracy, making it ideal for environments with limited computational resources like Colab or edge devices.**

**Its Limitations**

**Lower Accuracy Compared to Two-Stage Models – Faster R-CNN and YOLO may achieve higher detection accuracy, especially for small objects. While it may be fast, it may miss some objects or make lower confidence predictions compared to the heavier models.**

1. **Code Interpretation:**

* **Describe the role of the find\_images\_with\_classes function. Why is it useful when working with a large dataset like COCO?**

**Ans**

**The find\_images\_with\_classes function is designed to collect images that contain specific object classes from a large dataset like COCO or Pascal VOC.**

* **In the plot\_detections function, how does the threshold value (threshold=0.5) impact the number of objects displayed?**

**Ans**

**The threshold value 0.5 in plot\_detections controls which detected objects are displayed based on the model’s confidence score. If the threshold is too high, some valid detections may be missed. If the threshold is too low, the model might display incorrect objects (false positives).**

* **Explain how the heatmap visualization helps you understand the model's confidence in its detections.**

**Ans**

**A heatmap in object detection highlights the areas where the model is most confident in detecting objects. It overlays color intensity onto the image, helping to visualize how the model perceives objects. The heat map shows model focus, model strength, reveals model weakness and supports model improvement.**

1. **Observing Results and Limitations:**

* **Run the exercise multiple times. Which types of objects does the model tend to detect more accurately? Which ones are more challenging? Can you explain why?**

**Ans**

**The SSD detects larger and clearer objects but smaller or crowded objects may be challenging for the model due to reduced precision.**

* **Observe the bounding boxes. Are there any instances where the boxes are inaccurate or miss the object entirely? What factors in the images might be contributing to these errors?**

**Ans**

**Yes there are instances where the objects in the image are incorrectly identified. For example, misclassifying objects and making uncertain predictions like 0.50 for “chair” in the second image of the cars.**

* **How would you expect the accuracy of the model to change if we had used the entire Pascal VOC 2007 dataset instead of a small subset? Why?**

**Ans**

**This would most likely be of benefit to the model because a larger dataset helps the model learn more variations of each object (different angles and lighting) making it better at generalizing to new images.**

1. **Critical Thinking:**

* **How could you modify the code to detect a specific set of objects, like only animals or only vehicles?**

**Ans**

**To modify the code to detect only a specific set of objects, like only animals or only vehicles, I can apply class filtering by modifying the object detection pipeline.**

* **If you wanted to train your own object detection model, what steps would you need to take? What are some challenges you might encounter?**

**Ans**

**If I were to train my own model, I would define the problem or decide what objects I would like to detect, Collect and process the necessary data, train the model well, check its performance and then implement the model.**

**Challenges I might face would be; data quality & annotation errors, computational requirements, obstructions and small objects.**

* **Given the limitations of this model, in what real-world scenarios might it still be useful for object detection?**

**Ans**

**I believe it could still be used for wildlife monitoring, like detecting animals in camera trap footage for population studies. It may also help with identifying poachers or illegal activities in protected areas.**

**Erick Banegas**

* **My journey into object detection and machine learning has been both challenging and rewarding, filled with moments of frustration and triumph. From installing TensorFlow and managing dependencies to downloading datasets and loading pre-trained models, each step deepened my understanding of AI systems. Debugging issues, especially with missing libraries like kaggle hub, tested my patience but also sharpened my problem-solving skills. Seeing bounding boxes appear on images for the first time was a breakthrough moment, proving that all the effort had paid off. Evaluating the model’s accuracy taught me that AI isn’t just about making predictions—it’s about making precise and reliable ones. The real highlight was uploading my own images and watching the model process them, making everything feel more tangible and personal. This experience reinforced that persistence is key, debugging is an essential skill, and every challenge is an opportunity to learn. More than just writing code, I’ve begun to grasp the deeper intricacies of AI, and I’m excited to keep pushing my limits in this ever-evolving field.**

**NancyChieu**

**This assignment guides us into utilizing transfer learning as a part of deep learning to do object detection. The instruction for the assignment is very helpful to run the Google Colab to do object detection in Colab. Compared to our previous assignments, this assignment is easier and faster. By following the instruction to download Student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning.ipynb" and ran it in Google Colab, I was able to complete the first step without any issue and saved the lab result by changing runtime type and choosing the hardware accelerator as T4GPU. I went back to do the final step by re-open Collab again another day. It was smooth sailing until I was ready for final object detection processing. The suggestions for correction led me to redo many steps that I did. Finally I was able to complete the whole object detection. It was a valuable learning process to do trial and error to correct the mistakes and complete the lab work. My team leader, Matthew, helped me a lot with my learning.**

**Yoana Cook**

* **This lab provided a hands-on introduction to object detection using TensorFlow and the Pascal VOC 2007 dataset. Through this exercise, I gained a clearer understanding of the key differences between image classification and object detection. While image classification focuses on identifying what objects are present in an image, object detection goes a step further by also locating those objects with bounding boxes. This distinction became evident as I worked through the lab, visualizing both ground truth and predicted bounding boxes on the images.**
* **One of the main takeaways was the importance of using pre-trained models, such as SSD MobileNet V2, especially when working with limited computational resources. This model strikes a balance between efficiency and accuracy, making it suitable for tasks where computational power is constrained. However, I also learned that this efficiency comes with trade-offs, such as potentially lower accuracy compared to more complex models. This highlighted the need to choose the right tool for the task at hand, considering both performance and resource limitations.**
* **The lab also emphasized the significance of evaluation metrics in object detection, particularly Intersection over Union (IoU). By calculating IoU, I was able to assess how well the model's predicted bounding boxes aligned with the ground truth. This metric, along with precision and recall, provided a comprehensive view of the model's performance. It was interesting to see how adjusting the confidence threshold could impact these metrics, illustrating the trade-off between precision (fewer false positives) and recall (fewer missed objects).**
* **Visualizing the results was another key aspect of the lab. By plotting bounding boxes and class labels on the images, I could intuitively understand the model's strengths and weaknesses. For instance, the model tended to perform well on larger, more distinct objects but struggled with smaller or partially obscured ones. This observation underscored the challenges object detection models face in real-world scenarios, where objects can vary greatly in size, orientation, and visibility.**

**GitHub Link**

**Brooke Broderick**

**Matthew Choo**

[**https://github.com/MattChoo0/Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning-1-**](https://github.com/MattChoo0/Lab9VOC2007__Dataset_student_Notebook_LAB_Object_Detection_transfer_learning-1-)

**Cameroun White**

[**https://github.com/OhMyGodCam/hello-world/blob/main/Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning\_(1)\_(1)\_Copy.ipynb**](https://github.com/OhMyGodCam/hello-world/blob/main/Lab9VOC2007__Dataset_student_Notebook_LAB_Object_Detection_transfer_learning_(1)_(1)_Copy.ipynb)

**Melvis Maduagwu**

[**https://github.com/Melvis-07/Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_ Detection\_transfer\_learning-1-**](https://github.com/Melvis-07/Lab9VOC2007__Dataset_student_Notebook_LAB_Object_Detection_transfer_learning-1-)

**Erick Banegas**

[**https://github.com/erickxllx/jupyter-exploration**](https://github.com/erickxllx/jupyter-exploration)

**NancyChieu**

**https://github.com/nchieu2025/Lab9VOC2007\_\_Dataset\_student\_Notebook\_LAB\_Object\_Detection\_transfer\_learning\_(1).ipynb**

**Yoana Cook**

[**https://colab.research.google.com/drive/1t2kqGIKQOtSDEuku3ktQ3pIADj5CGOP\_#scrollTo=oMokp0C\_7NJb**](https://colab.research.google.com/drive/1t2kqGIKQOtSDEuku3ktQ3pIADj5CGOP_#scrollTo=oMokp0C_7NJb)