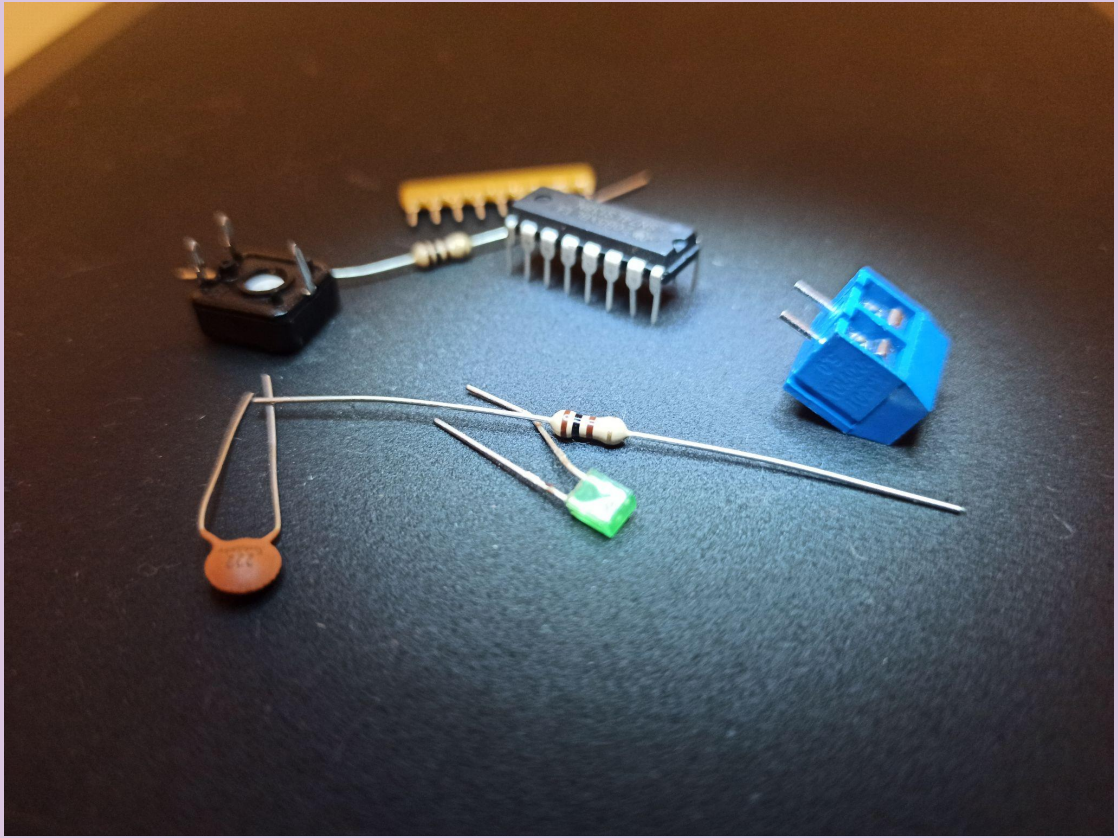
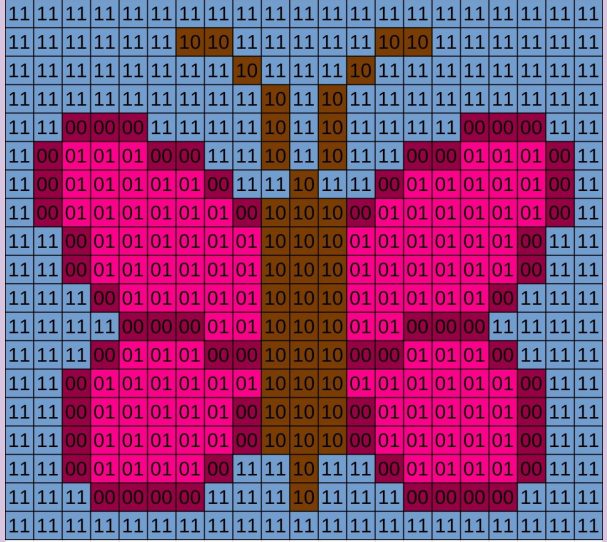
AI based Electronic Component Identifier

Student: Violet Concordia
Student number: B00125142
Supervisor: Benjamin Toland
Course ID: TU807

Object detection

- **Important in**
  - **Security**
    - **Notifying concerns**
      - People
      - Animals
      - Flora
  - **Production**
    - **Discarding defects**
      - Damage
  - **Analysis**
    - **Quality inspection**
      - Scratches
      - Spots
    - **Classification**
      - Resistor
      - Capacitor
    - **Total and class count**





11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 10 10 11 11 11 11 11 10 10 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 10 11 11 11 10 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 10 11 10 11 11 11 11 11 11 11 11 11 11
11 11 00 00 00 11 11 11 11 10 11 10 11 11 11 11 00 00 00 11 11 11
11 00 01 01 01 00 00 11 11 10 11 10 11 11 00 00 01 01 01 00 11 11
11 00 01 01 01 01 01 00 11 11 10 11 11 00 01 01 01 01 01 00 11 11
11 00 01 01 01 01 01 01 00 10 10 10 00 01 01 01 01 01 01 00 11 11
11 11 00 01 01 01 01 01 01 10 10 10 01 01 01 01 01 01 00 11 11 11
11 11 00 01 01 01 01 01 01 10 10 10 01 01 01 01 01 01 00 11 11 11
11 11 11 00 01 01 01 01 01 10 10 10 01 01 01 01 01 00 11 11 11 11
11 11 11 11 00 00 00 01 01 10 10 10 01 01 00 00 00 11 11 11 11 11
11 11 11 00 01 01 01 00 00 10 10 10 00 00 01 01 01 00 11 11 11 11
11 11 00 01 01 01 01 01 01 10 10 10 01 01 01 01 01 01 00 11 11 11
11 11 00 01 01 01 01 01 01 00 10 10 00 01 01 01 01 01 01 00 11 11
11 11 00 01 01 01 01 01 00 10 10 10 00 01 01 01 01 01 00 11 11 11
11 11 00 01 01 01 01 00 11 11 10 11 11 00 01 01 01 01 00 11 11 11
11 11 11 00 00 00 00 11 11 11 10 11 11 11 00 00 00 00 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11

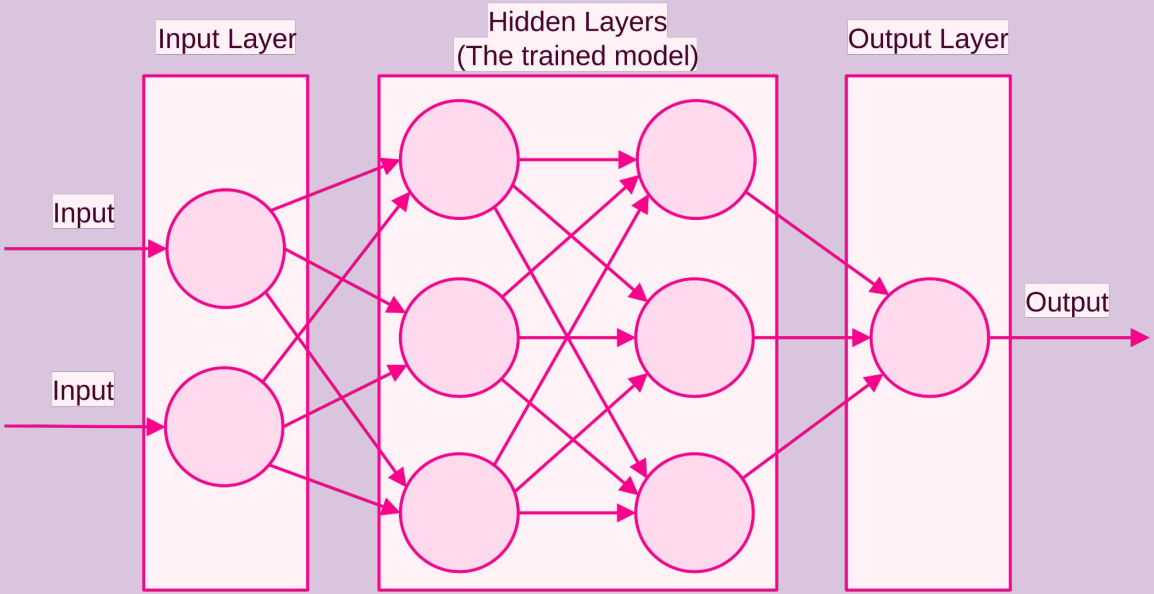
The problem

- **Object detection** from **images**:
  - Natural to **intelligent** creatures.
    - Designed for **object detection** through evolution.
      - Second nature.
  - **Binary data** to **computers**.
    - **Has no concept** of **object, image, or color**.
      - Everything is processed the same.

The solution

- **Neural Networks**
  - Complex networks of **neurons**.
  - Each **neuron** holds weight for specific input bias.
    - Trained against certain **datasets**, in hope of recreating the function.
  - The **output** is processed by the **neurons**.
- In image **object detection**, input is each individual **pixel**.

Simple Neural Network Diagram

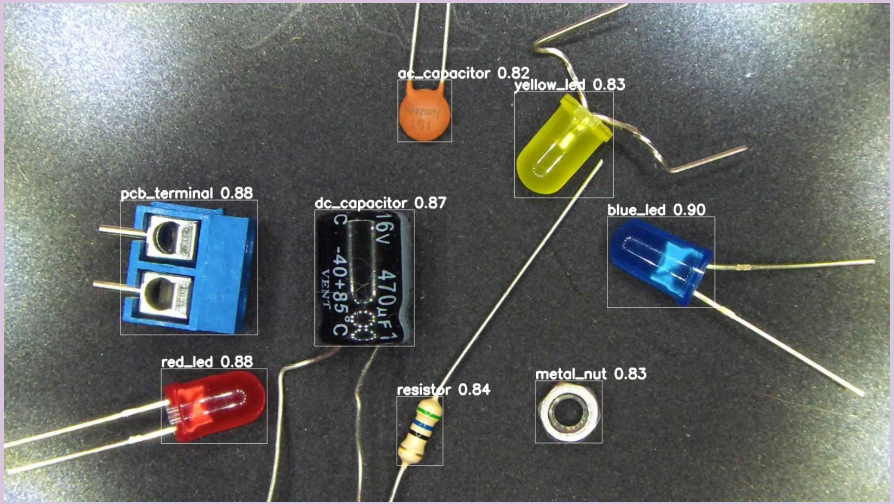


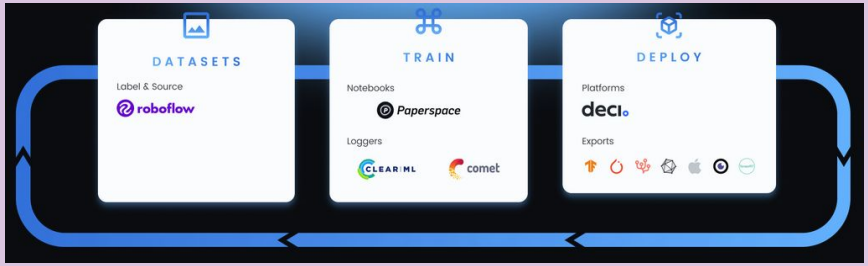
Chosen architecture

AI based Electronic Component Identification

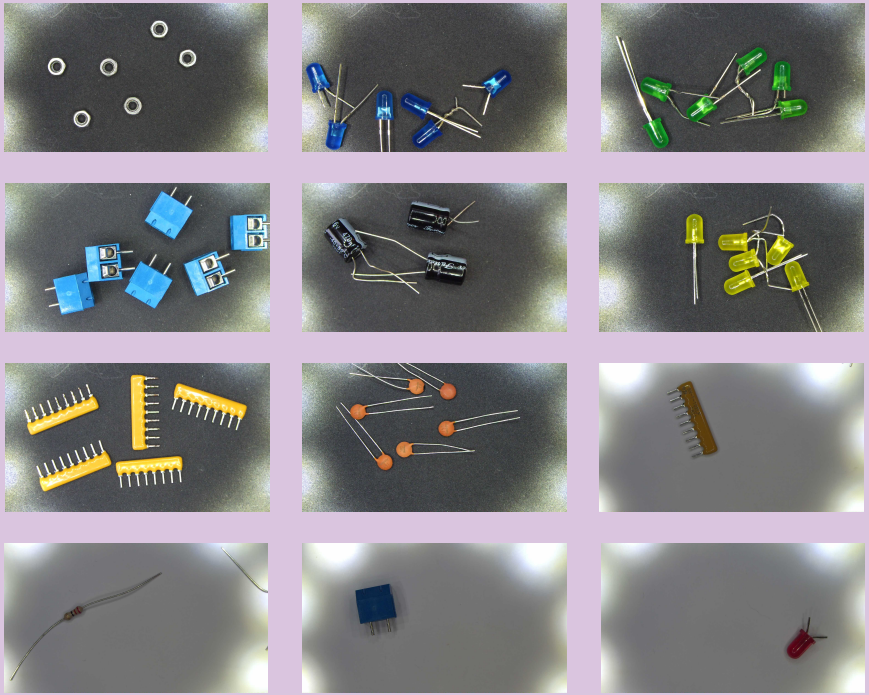
GUI display

- **Identifies**:
  - **Position** (Bounding Box on a live display).
  - **Class** name.
    - Resistor
    - Capacitor
  - **Confidence** (Percentage).
  - **Quantity**.
  - Potential **extra information**:
    - Resistors - **Color code** to Ohms.
    - Capacitors - **Number code** to farad.
    - IC - **Pin count**, **text** on the IC if visible.





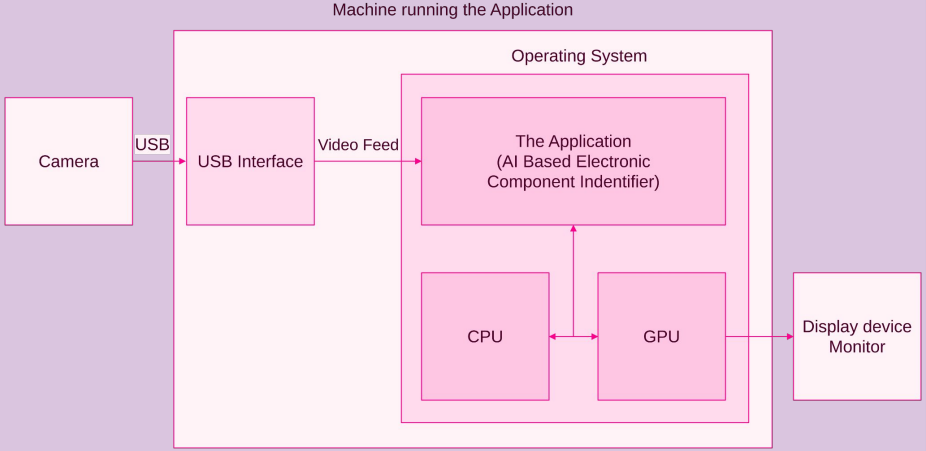
[1]



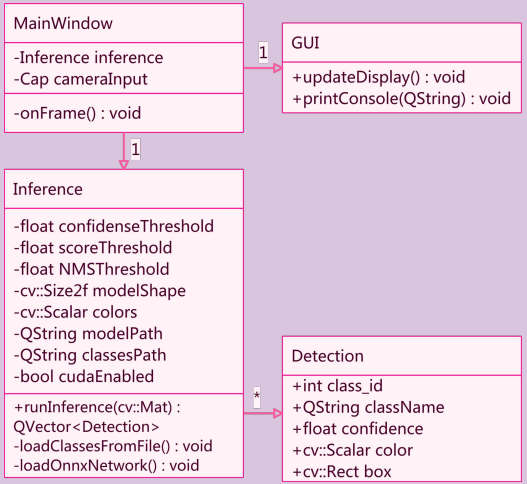
Model Training

- Trained on over **3000** images.
  - Taken on the **rig**.
  - Manually **labeled**.
  - **Classes**:
    - Resistor (Single, SIP)
    - Capacitor (AC, DC)
    - LED (Red, Green, Blue, Yellow, Clear)
    - Metal Nut
    - Integrated Circuits
    - Light Dependant Resistor
    - Diode
- **Each picture** should have some **variation** in conditions.
  - **Position**
  - **Rotation**
  - **Lighting**

System Block Diagram

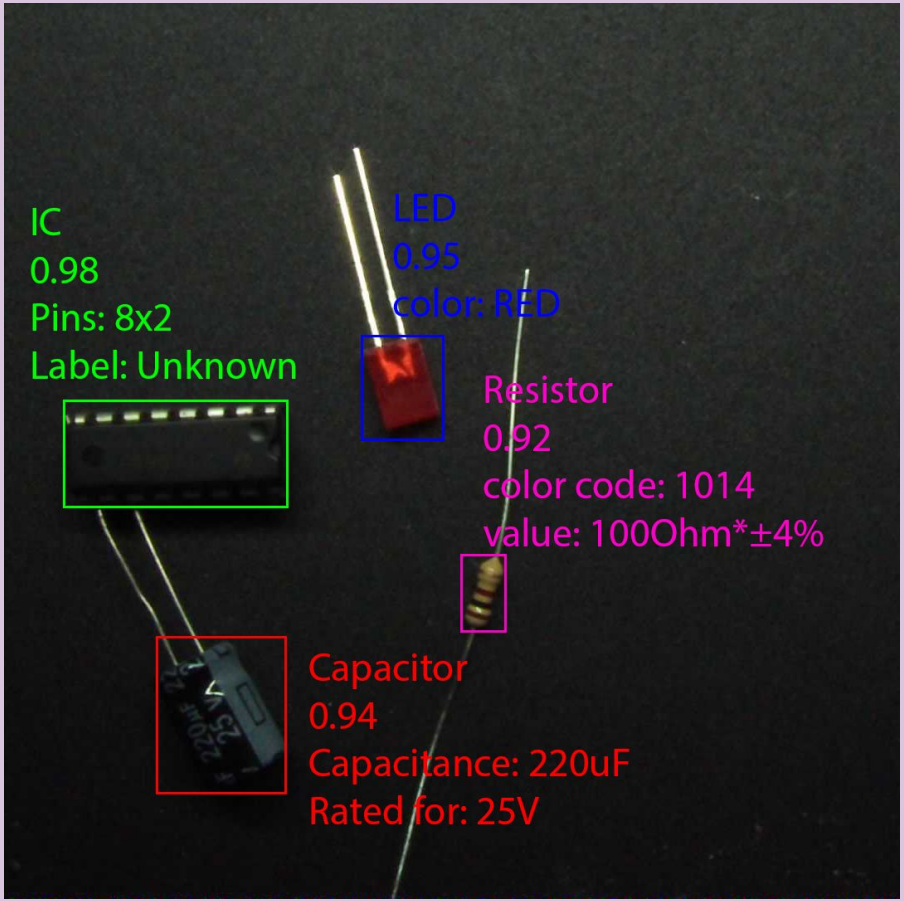


Class Diagram



Concept Diagrams

- System Block Diagram
  - **Camera** is External.
  - Communication through **USB**.
  - The application utilises both **CPU** and **GPU**.
- Class Diagram
  - MainWindow is the controller.
  - **Inference** is responsible for **object detection**.
  - **GUI** is the output.



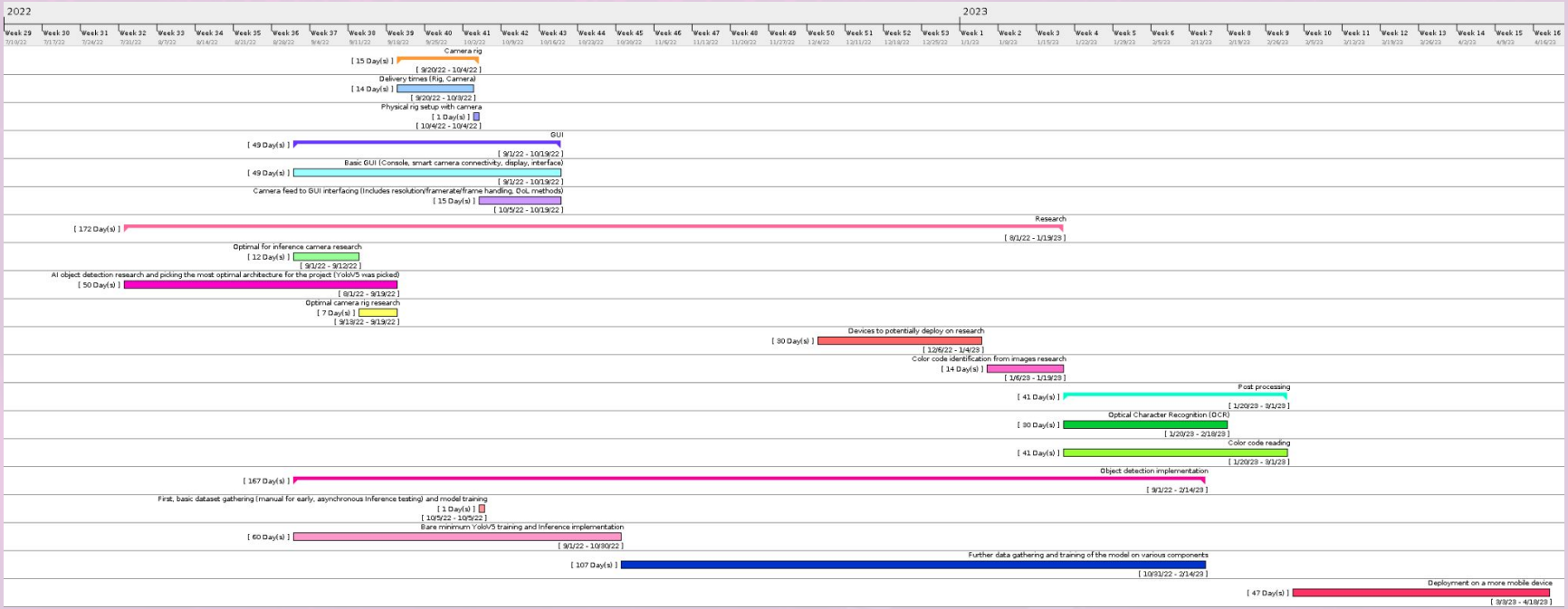
Note: For now, this was achieved through image manipulation.

Post-Processing

Additional processing of the **output** provided by the **inference**.

- **Identification** of (For those that apply)
  - **Labels** using **Text Recognition**
  - **Color/number codes**
  - **Pin count**
  - **Color** (LED)

Timetable


2022
2023
Week 29 7/10/22
Week 30 7/17/22
Week 31 7/24/22
Week 32 7/31/22
Week 33 8/7/22
Week 34 8/14/22
Week 35 8/21/22
Week 36 8/28/22
Week 37 9/4/22
Week 38 9/11/22
Week 39 9/18/22
Week 40 9/25/22
Week 41 10/2/22
Week 42 10/9/22
Week 43 10/16/22
Week 44 10/23/22
Week 45 10/30/22
Week 46 11/6/22
Week 47 11/13/22
Week 48 11/20/22
Week 49 11/27/22
Week 50 12/4/22
Week 51 12/11/22
Week 52 12/18/22
Week 53 12/25/22
Week 1 1/1/23
Week 2 1/8/23
Week 3 1/15/23
Week 4 1/22/23
Week 5 1/29/23
Week 6 2/5/23
Week 7 2/12/23
Week 8 2/19/23
Week 9 2/26/23
Week 10 3/5/23
Week 11 3/12/23
Week 12 3/19/23
Week 13 3/26/23
Week 14 4/2/23
Week 15 4/9/23
Week 16 4/16/23
Camera rig
[15 Day(s)]
[9/20/22 - 10/4/22]
Delivery times (Rig, Camera)
[14 Day(s)]
[9/20/22 - 10/3/22]
Physical rig setup with camera
[1 Day(s)]
[10/4/22 - 10/4/22]
GUI
[49 Day(s)]
[9/1/22 - 10/19/22]
Basic GUI (Console, smart camera connectivity, display, interface)
[49 Day(s)]
[9/1/22 - 10/19/22]
Camera feed to GUI interfacing (Includes resolution/framerate/frame handling, DoI, methods)
[15 Day(s)]
[10/5/22 - 10/19/22]
Research
[172 Day(s)]
[8/1/22 - 1/19/23]
Optimal for inference camera research
[12 Day(s)]
[9/1/22 - 9/12/22]
AI object detection research and picking the most optimal architecture for the project (YoloV5 was picked)
[50 Day(s)]
[8/1/22 - 9/19/22]
Optimal camera rig research
[7 Day(s)]
[9/13/22 - 9/19/22]
Devices to potentially deploy on research
[30 Day(s)]
[12/6/22 - 1/4/23]
Color code identification from images research
[14 Day(s)]
[1/6/23 - 1/19/23]
Post processing
[41 Day(s)]
[1/20/23 - 3/1/23]
Optical Character Recognition (OCR)
[30 Day(s)]
[1/20/23 - 2/18/23]
Color code reading
[41 Day(s)]
[1/20/23 - 3/1/23]
Object detection implementation
[167 Day(s)]
[9/1/22 - 2/14/23]
First, basic dataset gathering (manual for early, asynchronous Inference testing) and model training
[1 Day(s)]
[10/5/22 - 10/5/22]
Bare minimum YoloV5 training and Inference implementation
[60 Day(s)]
[9/1/22 - 10/30/22]
Further data gathering and training of the model on various components
[107 Day(s)]
[10/31/22 - 2/14/23]
Deployment on a more mobile device
[47 Day(s)]
[3/3/23 - 4/18/23]

[2]

Discussion

- **In comparison** to **algorithms**, and sometimes even **humans** - **inference** is capable of extracting an **Extremely high** level of detail from an input image.
  - As **post-processing** is **algorithmic**,
    it is expected to be **considerably less accurate**.
    - **Missed letters** must depend on **auto-correction**,
      which is another **algorithm**.
    - **Color shift** may result in a **wrong color code read**.
- **Gathered** over **3000 labeled images**.
  - The **training** of the current **model** for **300 epochs** takes over **10 hours**.

Conclusion

- Sufficient amount of **data** has now been **gathered**, yielding **impressive results**.
  - While the **project** is mostly **software focused**, a **high quality rig setup** was essential in gathering a **high quality dataset**.
  - This **concludes** the **gathering** of additional **data** to prioritise **other milestones**.
- The project is coming along great so far!
  - The next focus will be the **post-processing**.

References

- [1] YoloV5 https://github.com/ultralytics/yolov5, accessed on 6th of November, 2022
- [2] Gantt Project https://www.ganttproject.biz/, accessed on 6th of February, 2023

The end

Any questions?