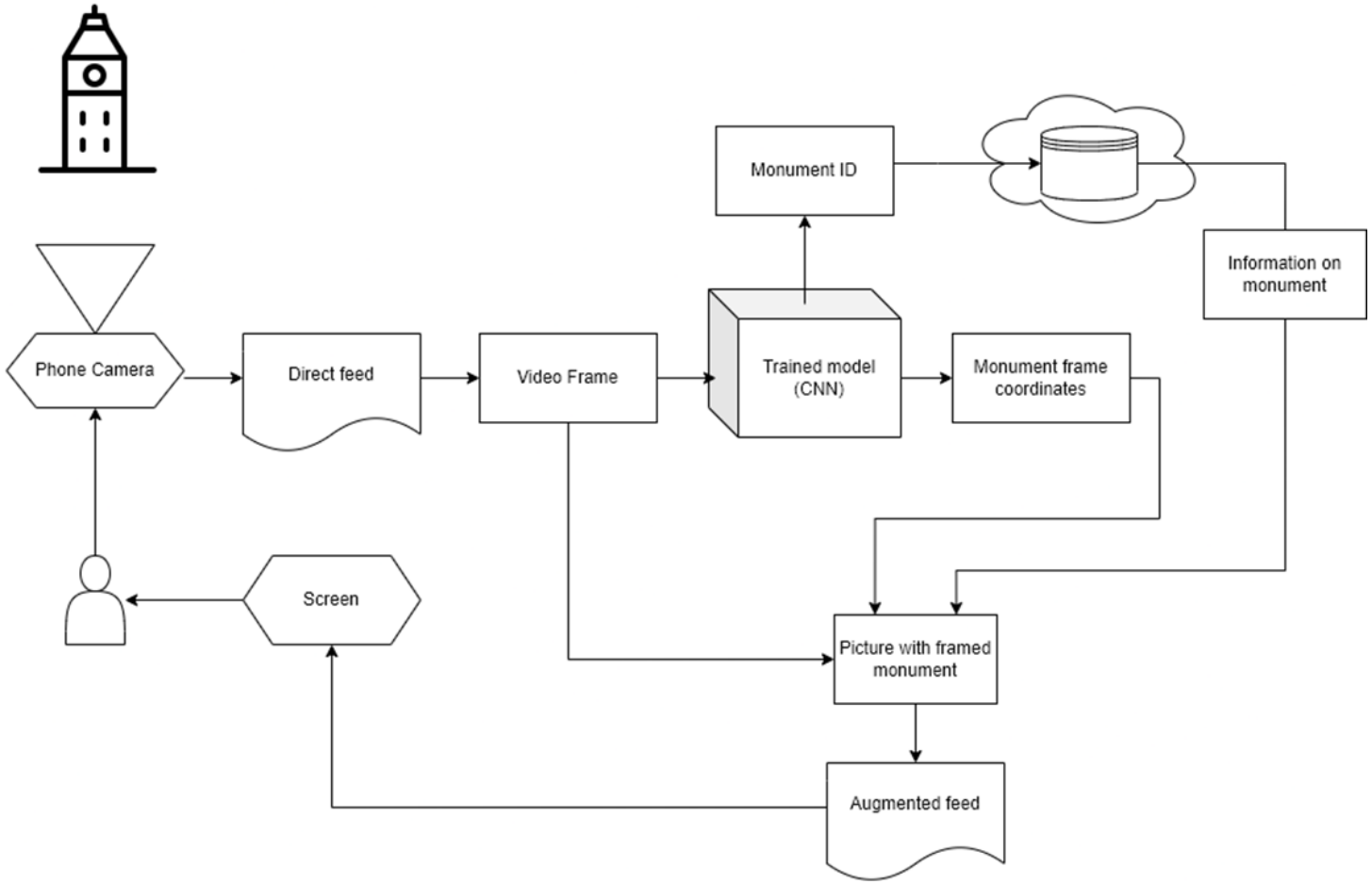
Deep Tour

Midterm review

Review

- Deep learning for tourism
- Augmented experience
- Providing information on landmarks
- Respect of privacy

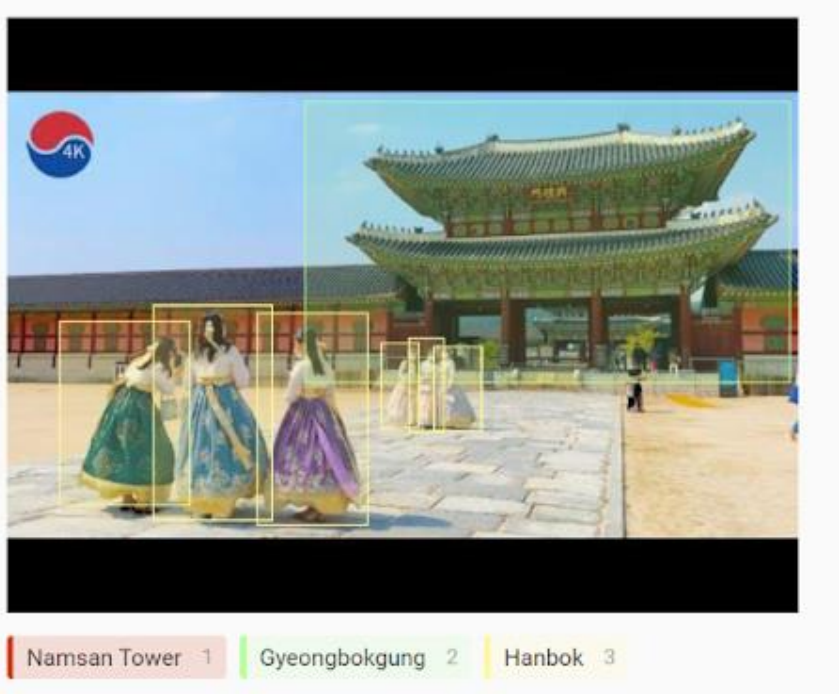


Data

- Focus on Seoul as a first prototype
- Google landmarks provides few pictures for Korean monuments
- Most links are outdated
- Scraping Google Images
- Manual labelling using label-studio
- Need to rethink scalability
- Progressing with a reduced database yet

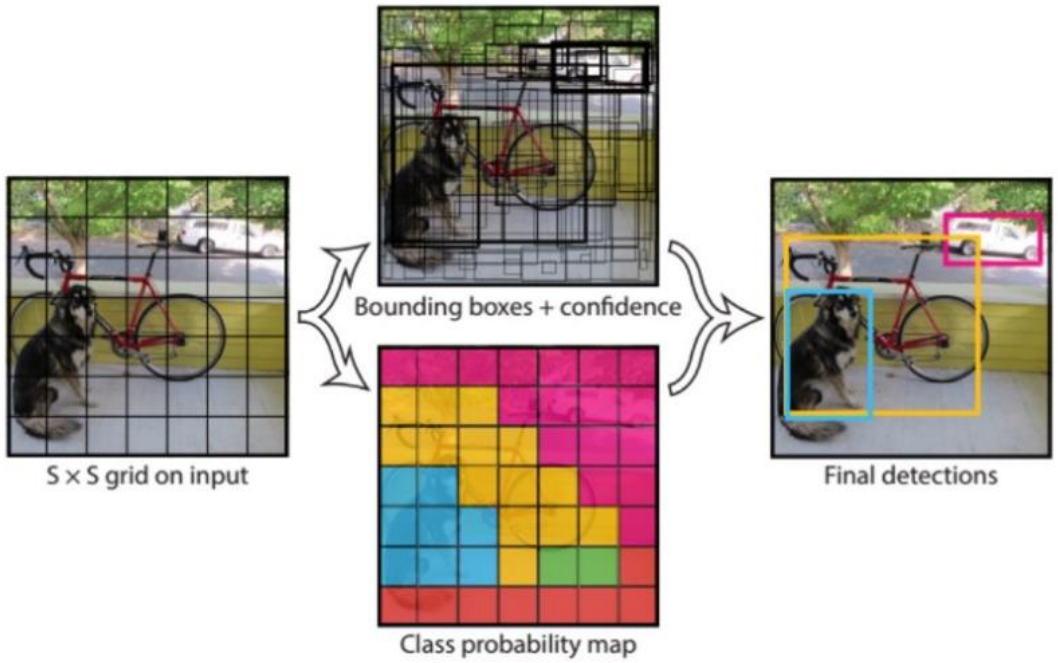
Data

- How to label huge monuments like Gyeonbokgung
- How to differentiate temples (similar color pattern and architecture)
- Participative data sourcing



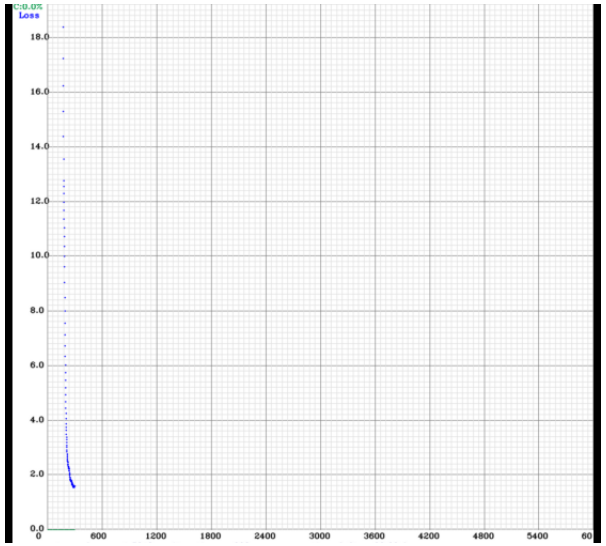
YOLO

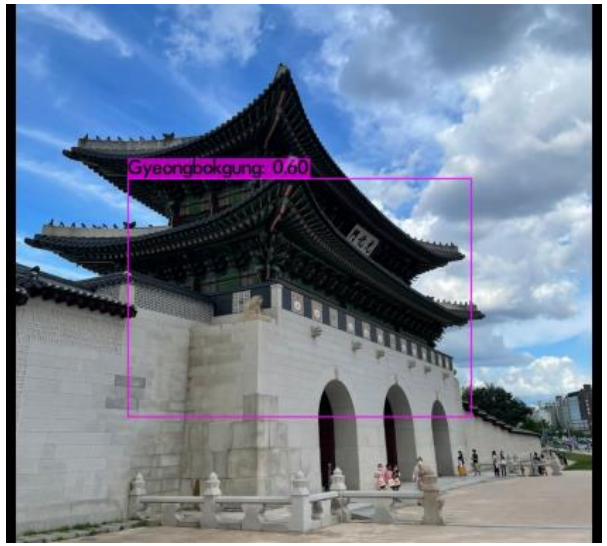
- High-level idea:
  - pass the image through CNN, splitting the image into m*m grid and generating for each grid bounding box and class probability
- Benefits:
  - Fast computation
  - Reasons globally about the image
  - learns generalizable representations of objects

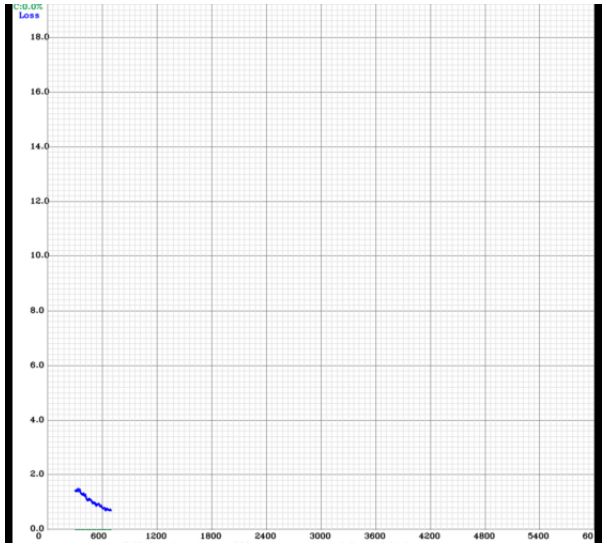


Demo

- Test at 300 and 700 epochs of training
- 3 Labels
- 500 pictures







Demo

- Threshold at 10% and 20%
- Efficient for single monuments
- Potent for multiple objects recognition but lower threshold

- Existing solution for object recognition
- Official example from TensorFlow Github

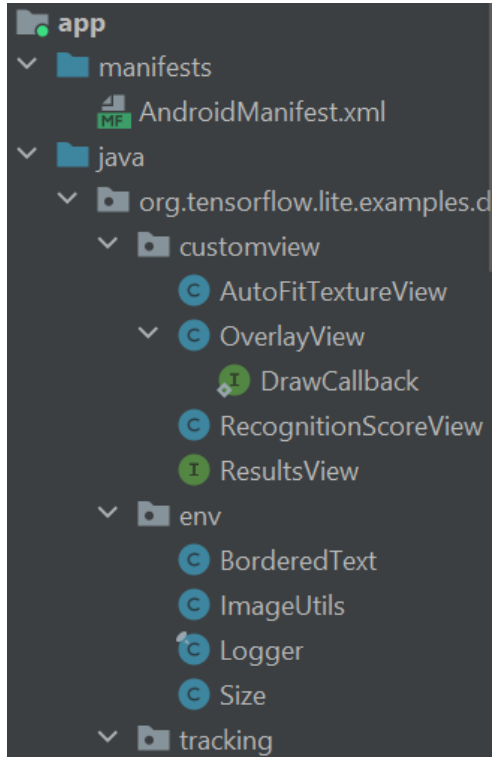
TensorFlow Lite Object Detection Android Demo

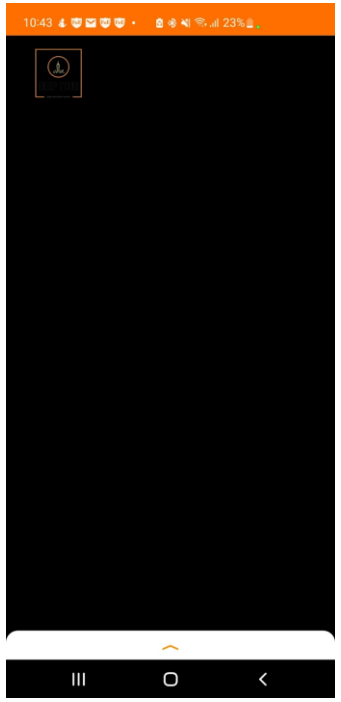
Overview

This is a camera app that continuously detects the objects (bounding boxes and classes) in the frames seen by your device's back camera, using a quantized MobileNet SSD model trained on the COCO dataset. These instructions walk you through building and running the demo on an Android device.

Done

- Quickly assimilate the application structure.
- Understand how to replace the existing model (.tflite format)
- Personalize the User Interface layouts. (logo…)





To Do

1. Merge the application and our model.
2. Add click button on recognized monuments leading to their information page.
3. Use the monument ID to fill each page from database.
4. Add optional features (Screen guidance…)

Model used

Downloading, extraction and placing it in assets folder has been managed automatically by download.gradle.

If you explicitly want to download the model, you can download from here. Extract the zip to get the .tflite and label file.

Custom model used

	Tasks	7 April	14 April	21 April	28 April	2 May	9 May	16 May	23 May	30 May	6 June
	Search a dataset to train	YL, ND		Done							
	Process dataset			YL, AP		Delayed					
Deep Learning (Python)	Build/Adapt a Neural Network model			AP, ND		Done					
	Train the model				AP		Delayed				
	Adapt the model for smartphone					AP, ND					
App dev (Android, Java)	Setup retrieval of video flux			YL, AP		Done					
	First UI design				YL, ND	Done					
	Retrieve coordinate of the monument in the screen				YL, ND						
	Setup basic content generation (Layout on Android)				YL, ND						
	Intermediate presentation				Preparation						
	Feedback modifications from Intermediate report						YL, ND, AP				
	Final UI design						YL, ND				
	Evaluation/Test in real condition							YL, AP			
Optional	User based data sourcing								ND, AP		
	360 camera feed								AP, YL		
	Screen guidance								YL, ND		
	User study							ND, AP			
	Final presentation									Preparation	

What next ?

- Participative data sourcing
  - Task rewards
- Limitation to one country/city
  - Different neuron weights for each country/city
  - Add-ons available based on localization
- 360 camera feed
  - Must keep the same NN
  - Rescaling video stream