Tunisian Republic
Ministry of Information Technology and Communication
Ministry of higher education and scientific research
ESPRIT

Graduation project Report

Computer Vision for

Elaborated by: Safa chehimi

Work proposed and elaborated within:

DNA Global Analytics

DNA GA

DNA GA Supervisor: Mr. touzi Foued

University Year: 2020-2021

Executive summary

Digital Technology is having an undeniable impact on how we carry out our daily routines, and it has also changed the way we do business. It is a fact that digitization has created new skill sets which most companies need to maintain their competitive edge.

Within DNA Value in DNA Global Analytics Swiss, we have launched a new Intelligence solution, where we use machine learning in particular Computer Vision,Nvidia edge device, data aggregation and data visualization to create a customer's data gathering platform. These insights will empower the business to make strategic study on customers and target their advertisement based on the real-time age and gender demographics in the store.Therefore be used to influence buying behavior of consumers which leads to a more profitable business.

The goals of the Competitive Intelligence project is to provide a customer segmentation to evolve the know-how marketing segment. This will provide insights about which DNA GA clients need to know about their clientele to stay ahead in the market strategy. Moving forward we wish to expand the project scope to be able to generate more intelligent features and services to help our clients make the right decision to ensure their business evolution.

Keywords: *Computer Vision,people counting,Age and gender classification, FairMOT, CNN, Transfer Learning, Visualization*

Dedication

Acknowledgement

I would like to thank everyone who contributed to the success of my internship.

I would show my special and greatest appreciation to DNA Global Analytics for hosting me and for giving me valuable assignments during this internship.

I also thank the entire DNA GA' team for creating a positive and comfortable work environment. They were a precious help in the most delicate moments especially my supervisor Foued Touzi. I really appreciated his continuous support, patience, kindness, motivation and immense knowledge. I could not have imagined having a better adviser and mentor.

I would like to thank Mrs Sonia Mesbeh who didn't consider distance as difficulty to take time to improve my advancements, review my report and especially support me with her kind words and unconditional support.

My sincere thanks goes to all my teachers at ESPRIT for the quality of education and for providing the theoretical knowledge I needed during the internship.

Finally, I take this opportunity to acknowledge with much appreciation the jury members who have taken the time to evaluate my work while hoping that they find in this report the qualities of clarity and motivation that they expect.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

DNA GA Digital Neuro-marketing Algorithm - Global Analytics

AI Artificial Intelligence

ML Machine Learning

CV Computer Vision

MOT Multi Object Tracking

YOLO You Only Look Once

PIL Python Imaging Library

VM Virtual Machine

NumPy Number Python

SSH Secure Shell

SSD Single Shot Detector

HOG Histogram of Oriented Gradients

RTSP Real-Time Streaming Protocol

RoI Region of Interest

GPU Graphic Processing Unit

IoU Intersection over Union

AP Average Precision

mAP mean Average Precision

KPI Key Performance Indicator

Introduction

For years, retailers have experimented with various techniques to study customer behavior. However, because of a lack of suitable technology and incorrect behavioral models, large amounts of data that can only be considered less than perfect have accumulated with no actual purpose.
thankfully with the rise of Artificial intelligence and Machine learning, retail business are trusting this modern technology to provide a business opportunity to beat the competitors whom are using a more traditional approach.

To gain new customers, maintain client loyalty, and improve customer service, retailers are revamping their business tactics. Knowing the quantity and specifics of visitors depending on gender is one technique to demonstrate a retail shop's good market performance. Installing CCTV cameras in retail stores allows you to monitor the number of visits. CCTV was only employed for security purposes in the existing store system to monitor activity in the retail business.
But now with the evolution of computer vision we are witnessing a major resurgence of interest in how machines 'sees' and how it can be used to build products for consumers and businesses. Few examples of such applications in retail store are: Inventory management, Inventory management, Store layout improvement and Virtual mirrors and recommendation engines.
DNA GA team, within Computer Vision Projects, focus on helping their clients in logterm marketing planning and strategy formulation. Among their top challenges is to help the client take the right decision at the right time to ensure its company's economic growth. In this study, we attempt to use computer vision to help our client (Cora Supermarket France) identify its target customers by processing their cars' data and extracting insights from it. Ultimately "Cora" will be offered a custom computer vision solution for intelligent customer behavior analytics with features including real-time people tracking and counting once entering the store beside gender identification and age estimation of their customers.

The remainder of this report will be detailed as follows

The first chapter will contain the overall overview of the project, beginning with presenting the hosting organism, the market study, then giving the project description. Throughout the second chapter, we represent our work methodology followed by the state-of-art in chapter three. In the fourth chapter, we will focus on the design and architecture decisions in this application, both from a general perspective as well as the proposed solution where we will discuss the reasoning behind our approach. The final chapter will focus on the achieved results and the implementation of the solution along with a guide on the various technologies used and the dashboard features.

I

Project Context

I.1 Project Context

In this section, we're going to present the general context of the project. Then, we will provide a description of the company which proposed and supervised it. During the latter, we will introduce its different center of interests. Then, we will present the market study, the problematic and the work methodology in DNA GA. Finally, we'll give a brief description of the solution's overview.

I.1.1 General Context

This project is done as part of my academic process, within the context of graduation internship in order to obtain my national engineering diploma of Telecommunications from the Higher School of Communications of Tunis (ESPRIT). It was done with collaboration of DNA GA Switzerland.

I.1.2 Company description

In this subsection, we will present the company DNA GA and introduce its different work fields.

I.1.2.1 About DNA GA

Founded in 2006, ADN Agency is managed by Bilel Chérif (the current CEO of the group) and accompanies clients in graphic design, consulting and outdoor and digital advertising management. ADN Agency has become DNA Global Analytics to cope with the fusion of the physical and digital worlds. DNA comes from digital neuromarketing algorithms. Its vision is to deal with marketing in a scientific way. DNA Global Analytics (or DNA GA) is the only participant today who can create physical data and merge it with digital data into "phygital" data [**?**] enabling brands to have a 360-degree consumer vision.

"Today, with our tools, we develop complex mathematical models that simplify marketing decisions" -Nihal Mougamadou, DNA GA CRO-

Figure I.1: DNA logo

I.1.2.2 Centers of interest

DNA GA works across industries at the intersection of emerging technologies with societal trends. Therefore, it has different fields of expertise. Among which, we can cite:

- **Blockchain:** The transparency created by the fully adopted blockchain can solve the problem of trust, control and arbitrage in media purchasing and placement by tracking the actions and discards of middlemen, who are prone to mistakes and even fraud [**?**].
- **Artificial Intelligence:** The goal is to use data to help brands better understand customers and build a bridge between online and offline brand experiences. In order to achieve this ambitious goal, a unique method was constructed by merging online and offline data (such as billboards, advertising screens, navigation bars, surveillance cameras, etc.), and using machine learning algorithms to find patterns through large amounts of data [**?**].
- **Technology:** Manufacture smart cameras to analyze customer behavior in real time. The AI team then processes and analyzes the collected data to create a user-friendly dashboard that uses data visualization technology to display key information [**?**].
- **Sport and Entertainment:** Today, the sports and entertainment industries are being developed in the direction of complete "phygitalisation". Fans hope that their teams and their favorite players can have a better experience, and the new business model is becoming more and more mature. Industry leaders need to use new technologies to reinvent themselves in order to maintain their position [**?**].
- **University:** In addition to technical expertise, DNA also collaborates with several top

universities, renowned professors, researchers and business experts. Indeed, the commitment to the development of blockchain and artificial intelligence in multiple fields such as Marketing, Finance, Sports, Security and Safety has led to regular publication of research and participation in the creation of research papers.

- **Health and Safety:** Organizations must frequently cope with unexpected events such as the COVID-19 pandemic or ever-existing threats like terrorism. These "Black swan" events or perpetual threats to people safety represent new challenges and expectations regarding security as well as business challenges.

I.1.3 Market study

Gathering Data in retail utilizing computer vision is not a novelty.in fact Computer vision is the most technologically mature field in modern artificial intelligence and business skateholders are finding ways to utilize this technology across every sector of the economy, health, manufacturing and retail. One of these business is **VisionLabs** a dutch company specialized in AI applications mainly in computer vision, a filed that is also shared with our hosting company DNA.
Vision Labs proposes a solution similar to the one we had in mind. they offer products implementing efficient and cost-effective solutions for customer flow analysis in retail facilities of any size. their solution is a mobile or web application that need to be installed by the customer before entering the store. their products can provide this data while detecting:

- Age, gender and emotion classification.
- Visitor groups.
- Dwell and attention time
- Unique and returning visitors

A solution with various features that definitely inspire us in our work, but it lacks even more important features our client is searching for. One being counting people inside the store and two it need to be an end to end solution that uses the installed CCTV camera footage as the data source, a not a mobile application that its use depends on customer.So it now

clear to sat that this solution and like many more found on the internet doesn't not match the clients specific needs.

I.1.4 Problematic

Our client, a French hypermarket, wants to incorporate cutting-edge technology into their daily operations in order to provide outstanding customer service, improve decision-making, and gain a deeper understanding of their customers' interests. Cora opted for an AI-powered monitoring system to improve their strategy and obtain useful data.

Before us, Cora was adopting a traditional way of storing and processing data. Its main focus was to collect data and ensure clientele understanding through online shopping on the french website. The use of cookies and data collection within the browsers enables the marketing managers to make decisions about upcoming promotions, sales and even enhance their recommendation system.

However in the retail store acquiring the customer's data is a hard task as loyalty cards are often not used while purchasing, most of the time customers forget to bring it , lose it or just decide not to use it. So the customer's understanding depends solely on the fact if they have this loyalty card or not.

Furthermore, due to covid-19 restrictions, to minimize the number of people, Cora closes its store doors whenever the security agent feels like the store is a bit crowded, the decision is based on the man's perspective and not backed by real statistics which can be damaging both for people health if the number of people was higher than the one perceived to be and can be damaging to Cora business if security agent was letting less people than the maximum required .

Through our meeting with cora we noted it's business needs and functional requirements, they wanted to capture variable data on consumer behavior, people leaving and entering the store, the number of visitors for every minute, their demographic as in age and gender precisely, and it would be integrated with a camera system. A solution like this would provide our client with a broad perspective of their data, allowing them to make data-driven decisions, by combining all of these elements.

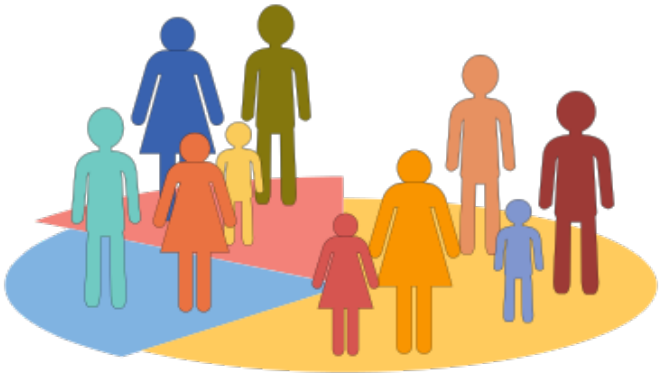


Figure I.2: Customers segments

I.1.5 Solution

On one hand The number of individuals that visit a store or a shopping complex provides important information about how well retail functions. This data can be used to estimate the number of visits, as well as the busiest days and hours. Furthermore, the data can be used to determine how shopper count patterns change over time, as well as the impact of promotions, advertising, competition, weather, school holidays, road closures, and other factors on visitor counts. It can also be used to assess the impact of relocating merchandise and concessions.

on the other hand One of the major aspects that can influence a customer's decision is gender. women and men approach shopping with different perspectives and considerations, women shop out of boredom and fun In contrast to men who shop out of necessity, women go on shopping to purchase both essential and non essential goods, to socialize and be stress free. In addition, female are easily attracted by advertisements compared to male. Therefore, gender identification is one of marketing techniques that can be used to influence buying behavior of consumers.
By combining the two major elements we can leverage data collection in retail though AI to ensure Cora's business growth and to adjust the company's marketing strategy.

To answer to the requirements of our client we had to:

- Decide on the most fitting hardware suited for all the technical requirements;

- Collect and analyze in-store CCTV camera footage , to classify the customers into their appropriate age and gender groups.
- Ensure accurate maximum people allowed in the store to answer to human COVID-19 preventive measures.
- the AI solution is real-time with minimal latency.
- Deploy the solution within a short time frame.

finally,We collaborated closely with the client's team to ensure that the solution was implemented smoothly and in accordance with the defined goals. From the initial environment evaluation to system configuration and data science model training.

I.1.5.1 Project Time Management

A project management methodology is a set of principles and practices that guide you in organizing your projects to ensure their optimum performance one of the six major functions is Time management which is the management of the time spent, and progress made, on project tasks and activities. Excellent time management requires the planning, scheduling, monitoring, and controlling of all project activities. As in all software projects , one is advised to choose the project management methodology that is best suited for the project. the methodology was chosen according to:

- **Flexibility:** outset what the solution will look like, project is prone to change, not sure how the solution outset will look like
- **Timeline:** client wants to be involved in the process and see results and progress , so work needs to be quick and efficient.

Our projects require extreme flexibility and speed. It is also iterative and incremental. It is best suited to the agile project management method because team members can rapidly adjust tasks as needed to ensure that the customer or business is always satisfied and is provided with outcomes that result in benefits.

Through this method, the project is broken down into "sprints" that are our three business questions.

Our Agile working process can be broken down into six steps.

1. The first step is to identify the project vision. It defines what the project is, how it will support the business strategy, who will benefit from it and how that will happen.
2. The second step is to build the project road-map where the requirements are set.
3. The third step is to create a release plan that consists in fixing the sprint's time-frame.
4. The fourth step is to plan and execute each sprint. The fifth step is to hold weekly meetings. These meetings should discuss what was done the previous week and what will be done that week.
5. The sixth step is to hold sprint reviews. These occur at the end of each sprint and consist of presenting the output to the data science team for feedback.

   Within the project there is a cycle that will be repeated and only end when the project is deemed finished.

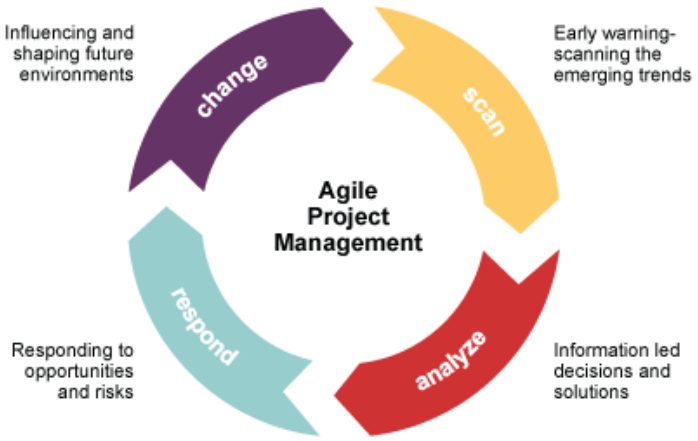


Figure I.3: Agile working cycle

I.1.5.2 Project Management Method

Since time is an important factor in the project, this work should involve precise project time management. To have an overview of the time taken for each phase, Figure below shows how time is partitioned.

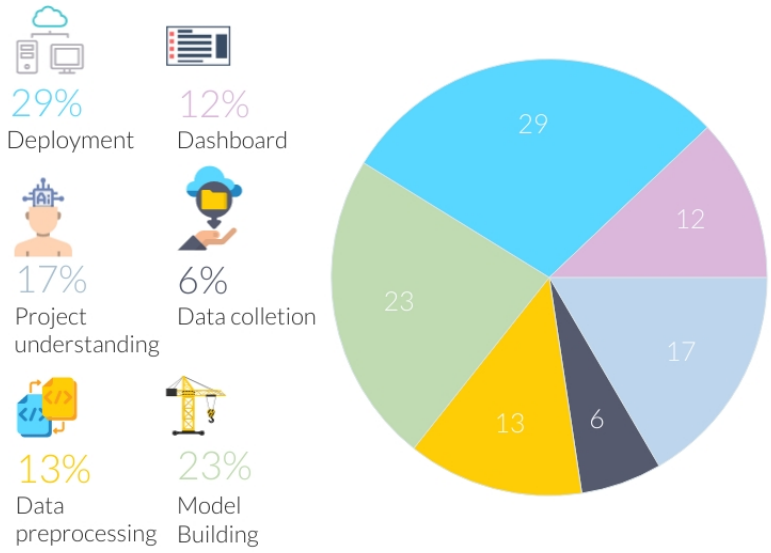


Figure I.4: Time Management

Most of time spent is spent deploying the project on the end device since we implemented advanced techniques in the process so that the models we've build run of full speed in real-time with excellent performance and low power consumption. Data preparation (Data collection, cleaning, pre-processing, labelling) accounts for about 20% of the work, we've spent this time in making data ready for training. we've dedicated 45 days almost 23% building the core of our project, the AI models themselves ,it took us numerous trials to be content with the results.

I.1.5.3 Gantt chart

In this subsection, we will use Gantt chart, which is a type of bar chart, to illustrate the project schedule. In this graph, we will list the tasks to be performed on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars in the graph represents the duration of each activity.
On this Gantt chart we can see [**?**]:

- The start date of the project.
- what the project main tasks and sub-tasks are.
- Starting date and the estimated time of arrival (ETA) of the tasks.

- The duration for every task in days.
- Notes and descriptions.

	Task Name	Duration	Start	ETA	Notes
1	Complete project execution	~200 days	15.02.2021	01.09.2021	
2	Project understanding and system design	33 days	15.02.2021	20.03.2021	
3	Project understanding and research	14 days	15.03.2021	29.03.2021	
4	system design	5 days	29.02.201	03.03.2021	architecture of the whole project
5	choose a cloud provider	2 days	03.03.2021	05.03.2021	AWS, GCP or Azure
6	Configure the Virtual machine from the cloud provider	2 days	05.03.2021	07.03.2021	GCP VM configiration
7	choose a cloud database and create a new project	2 days	07.03.2021	09.03.2021	MongoDB or firebase or other
8	Identifying the project vision and roadmap	9 day	09.03.2021	17.03.2021	roadmap -Gantt chart
9	Deep learning reasearch	15 days	05.03.2021	20.03.2021	
10	Data Collection	12 days	20.04.2021	01.04.2021	
11	Identifying the data sources	3 days	20.03.2021	23.03.2021	with the help of CRO and project manager
12	Collect data	4 days	23.3.2021	27.03.2021	Collect data from open source datasets
13	Upload data in google cloud VM	3 days	27.03.2021	30.03.2021	upload via SCP in command line
14	supervision and meetings	2 days	30.03.2021	01.04.2021	first restitution with madame Wissal Neji
15	Data preprocessing	26 days	01.04.2021	27.04.2021	
16	Data cleaning	10 days	27.04.2021	07.04.2021	outiliers-missing data
17	Data integration	4 days	07.04.2021	11.04.2021	combine multiple features in one
18	data reduction	4 days	11.04.2021	15.04.2021	data compression (images)
19	Data transformation	6 days	15.04.2021	25.04.2021	aggregation,normalization
20	data preprocesssing validation	2 days	25.04.2021	27.04.2021	validate this phase with the supervisor
21	Model Building	45 days	27.04.2021	11.06.2021	
22	Compare and choose object detection models	8 days	24.04.2021	06.05.2021	fairMOT, yolov3,SSD...
23	compare and choose tracking techniques	5 days	06.05.2021	11.05.2021	centroid tracker, JDETracker...
24	Compare and choose a CNN classification model	4 days	4.05.12	08.05.2021	vgg16,vgg16,resnet
25	Retrain FairMOT with age and gender on imdb dataset	12 days	08.05.2021	20.05.2021	retrain Fairmot to train for detection, reid,age, geder
26	Build age and gender models separately	15 days	08.05.2021	23.05.2021	tryning the models from one we compared
27	Try different CNN models	1 day	25.05.2021	26.05.2021	
28	code the direction classifier	3 day	04.05.2021	07.05.2021	python script using detection model and the track ids
29	Testing the models on real data	1 day	07.05.2021	08.05.2021	outside data from open source videos
30	test on real client data	2 days	08.05.2021	10.05.2021	client provided a video snippet
31	testing and comparing different models	1 day	10.05.2021	11.05.2021	
32	Deploy the solution on an edge device	58 days	11.06.2021	08.08.2021	
33	convert all models to onnx Create a TensorRT engine	11 days	13.06.2021	24.06.2021	(nano, xavier nx ,AGX) onnx converter in pytorch and TF.
34	Research GPU available edges devices	4 days	24.06.2021	08.06.2021	inside the jetson device using CUDA
35	Run inference from the TensorRT	4 days	28.06.2021	30.06.2021	infere the models
36	Implement the models and add threads in the main script	2 days	30.06.2021	1.07.2021	threading library in python3
37	Implement the gstreamer I/O	6 days	01.07.2021	07.07.2021	camera input and video strea output configuration
38	Configure the local network: ssh and rtsp ports	3 dyas	07.07.2021	10.07.2021	configure the ports and protocols used in the streaming
39	broadcast the CCTV xamera Stream	1 day	10.07.2021	11.07.2021	using gst streamer 1.0
40	Create mongodb cloud cluster and the connection string	5 days	11.07.2021	16.07.2021	mongodb atlas on AWS provider
41	Connect to the cluster from the edge device	15 days	16.07.2021	08.08.2021	ion the jetson: using pymongo , mongodb driver for python
42	Dashboard Visualization	24 days	08.08.2021	01-09-2021	
43	Retrieving the correspondant data in real-time	12 days	08.08.2021	20.08.2021	update the graph in real -time
44	Data visualization & Identifying the KPI's	12 days	20.08.2021	01.09.2021	implement usefull kpi's

Figure I.5: Table of the project tasks

After preparing the table of the project tasks, we translated it into Gantt Chart illustrating the evolution of each step of the work with the corresponding delays.

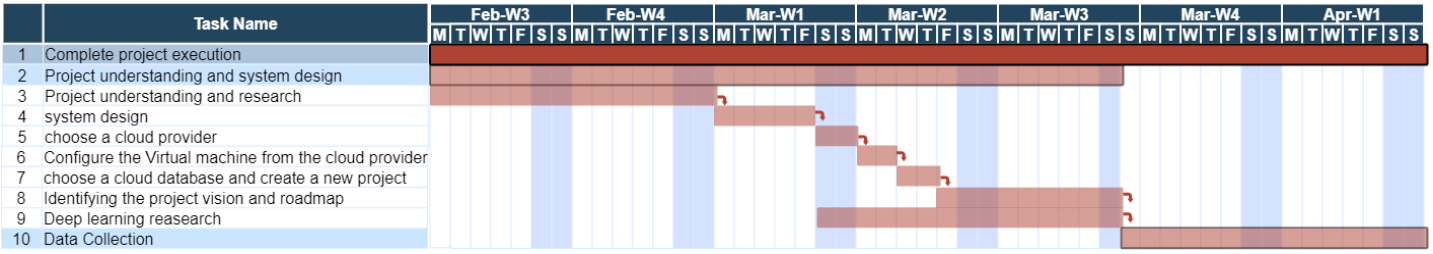


Figure I.6: Project understanding's Gantt Chart

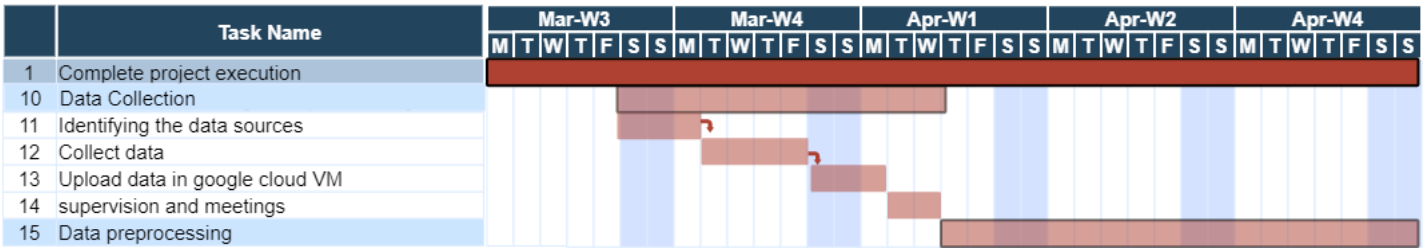


Figure I.7: data collection's Gantt Chart

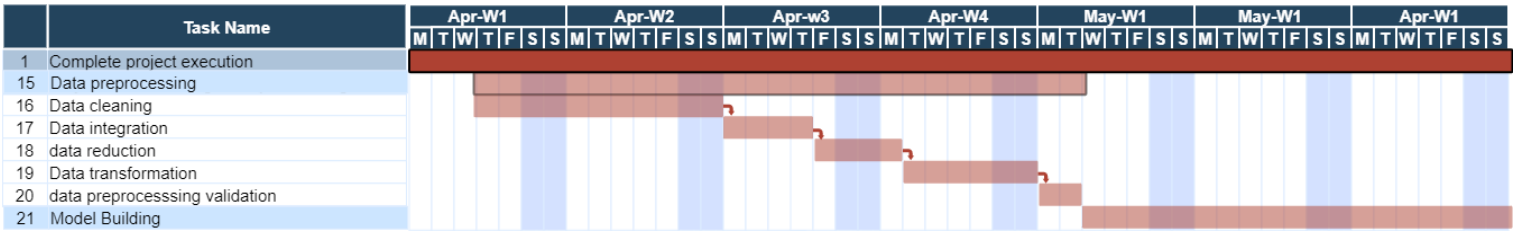


Figure I.8: data preprocessing's Gantt Chart

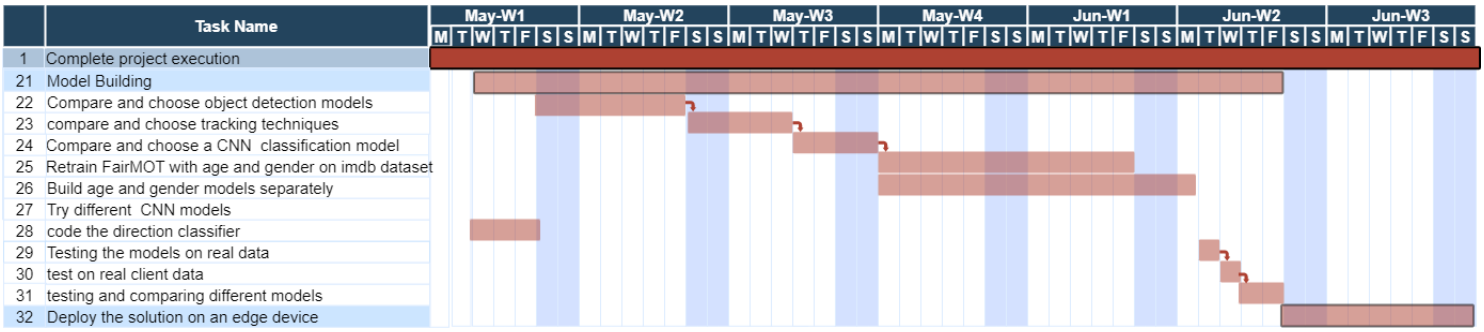


Figure I.9: Model building's Gantt Chart

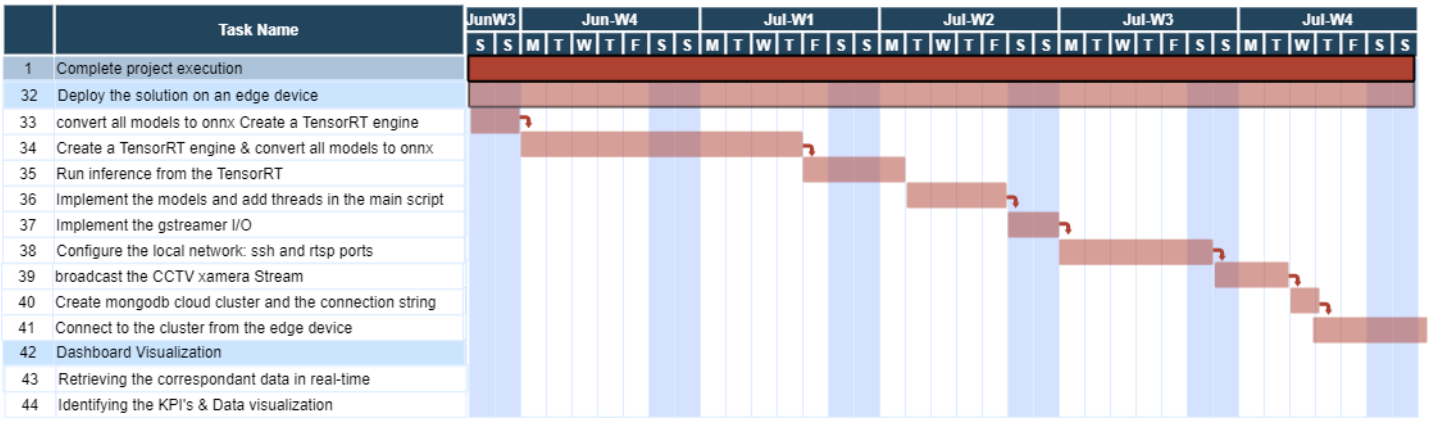


Figure I.10: solution deployement's Gantt Chart

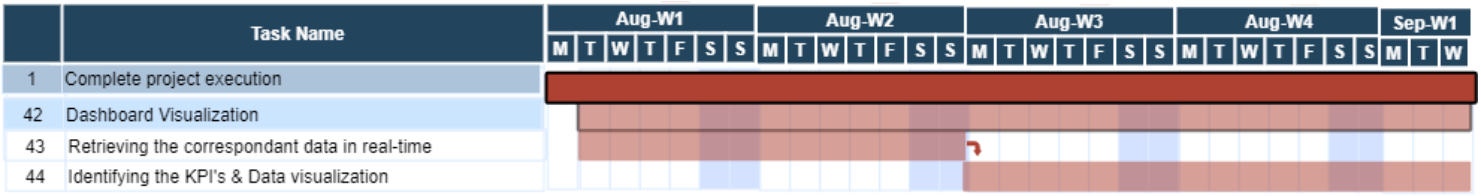


Figure I.11: dashboard visualization's Gantt Chart

II

Work Environment

Introduction

The project needs to answer to both requirements count people in the retail store and understand their demographic segmentation . For that reason, we ought to acquire a proper data-set to use in making the AI models responsible for the demographic segmentation alongside the tools needed to accomplish these project's requirements.

II.1 Data Sources

Before moving to the tools, we need to identify which data exactly is to extract for the age and gender classification task. we opted for an open source dataset from one of the many available online to train the model. Moreover to test the model on real production like imagery, we requested videos from Cora France CCTV security camera and we transformed them to ready to use test dataset following these steps.

- **From videos to frames**

  Cora provided us with a raw 30min video footage from the CCTV camera, we needed to convert that video into frames meaning a set of images. the video has a 60FPS rate if we extracted all the frames in the video we would have 1800*60 which equals to 108K images resulted from the transformation. We decided to extract frames based on 10FPS resulting in 18k images a number that is manageable for further transformations.

  FFMPEG a popular multimedia framework was used to generate the frames from the video.

- **From frames to cropped face images**

  To extract the faces from the frames we had to run the folder containing the images through a face detector.

  Face detection is an artificial intelligence used to find and identify human faces in digital images through a face detector. It uses algorithms and ML to find human faces within larger images, which often incorporate other non-face objects such as buildings, landscapes, and other human body parts like hands and feet. we went with a face detector that can be easily integrate in opencv librabry for ease of use, Dlib HoG was our detector of choice, the fastest method on CPU that can detect small images as

small as 70X70 pixels. a constraint that is fulfilled.

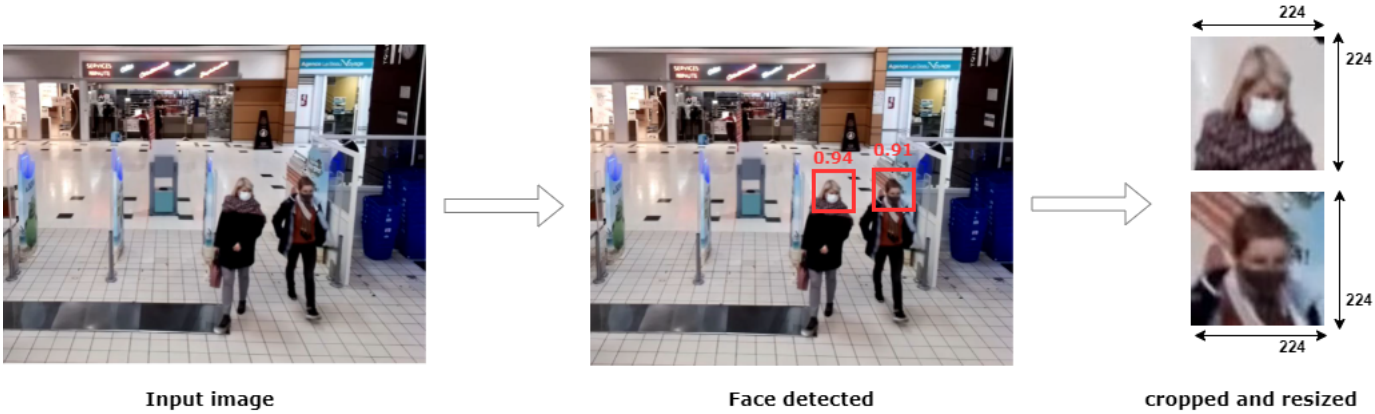


Figure II.1: face detection and cropping process

after cropping and resizing the images as 224X224 pixels a resolution that was not picked randomly but it was a well thought of decision we will discuss it further in the next chapters. the results were saved in a folder automatically.

- **Manual classification of age and gender**

  Since we want the additional test dataset to be as humanly accurate we manually created gender folders containing female and male images separately. And we did the same process for age.

  With this our additional real world dataset is ready for later use.

II.2 Prerequisites for data preparation

To prepare an accurate model, we need a huge data-set that will be trained.as known the biggest time consuming task of any Machine learning project is data cleaning and downloading the biggest demographic datasets isn't gonna make this task any less tidiess. for this we need the right ecosystem , i have chosen the following based on community availability, popularity and ease of use.

- **Anaconda:** Anaconda Individual Edition is the world's most popular Python distribution platform with over 20 million users worldwide. In which, we created different

virtual environments for different tasks (web-scraping, models training,...), each contains the appropriate packages, libraries and versions [?].

Figure II.2: Anaconda logo

Using the conda package manager, we installed the software "Jupyter Notebook" .

Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows us to create and share documents that contain live code, equations, visualizations and narrative text. It's useful for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, web-scraping, etc [?].

Figure II.3: Jupyter Notebook logo

Python 3: It is a programming language which has efficient high-level data structures and effective approach towards object-oriented programming. Python's elegant syntax and dynamic typing make it an ideal language for scripting and building ML models [?].

Figure II.4: Pyhton 3 logo

PIL Library: Python Imaging Library(PIL), in newer versions known as Pillow, is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats [?].we used this library in the face detection process detailed above

Figure II.5: PIL logo

II.2.1 Data Pre-Processing Tools

Now, that we had the required data, a crucial step is necessary for a goal-oriented work: Data pre-processing. Thus, before using that data for the purpose we want, we need it to be as organized and "clean" as possible. Therefore, we'd use the following techniques to get a better data-set.

- **Opencv:** OpenCV-Python is a library of Python bindings designed to solve CV problems. It supports a lot of algorithms related to Machine Learning and is expanding day-by-day.

Figure II.6: Opencv-Python logo

- **Pandas:** Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,built on top of the Python programming language. [**?**].

Figure II.7: Pandas logo

- **Numpy:** NumPy is a low level library written in C (and Fortran) for high level mathematical functions. NumPy cleverly overcomes the problem of running slower algorithms on Python by using multidimensional arrays and functions that operate on arrays. Any algorithm can then be expressed as a function on arrays, allowing the algorithms to be run quickly [**?**].

Figure II.8: Numpy logo

- **Matplotlib:** Matplotlib is a flexible plotting library for creating interactive 2D and 3D plots that can also be saved as manuscript-quality figures. The API in many ways reflects that of MATLAB, easing transition of MATLAB users to Python [**?**].

Figure II.9: Matplotlib logo

II.2.2 Models Building and deployment Tools

- **google cloud :** Compute Engine is a Secure and customizable compute service that lets you create and run virtual machines on Google's infrastructure.With VMs, we have total control over the configuration and can install anything we need to perform the work.

Figure II.10: google cloud logo

- **SSH:** The SSH protocol uses encryption to secure the connection between a client and a server. All user authentication, commands, output, and file transfers are encrypted to protect against attacks in the network. We have used this protocol to connect to the nvidia jetson xavier.

Figure II.11: SSH logo

- **Tensorflow:** TensorFlow is an end-to-end open source platform for ML. It has a comprehensive, flexible ecosystem of tools, libraries and community resources.

Figure II.12: Tensorflow logo

- **Keras:** Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow and is designed to enable fast experimentation with deep neural networks.

Figure II.13: Keras logo

- **ONNX:** ONNX is an open format built to represent machine learning models. ONNX defines a common set of operators - the building blocks of machine learning and deep learning models - and a common file format to enable AI developers to use models with a variety of frameworks, tools, runtimes, and compilers [**?**].

Figure II.14: ONNX logo

- **TensorRT:** NVIDIA® TensorRT™ is an SDK for high-performance deep learning inference. It includes a deep learning inference optimizer and runtime that delivers low latency and high throughput for deep learning inference applications [**?**].

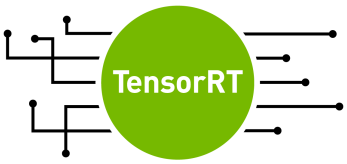


Figure II.15: TensorRT logo

II.2.3 Dashboard Building Tools

- **PyMongo:** PyMongo is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python. This documentation attempts to explain everything you need to know to use PyMongo. [**?**].

Figure II.16: PyMongo logo

- **MongoDB:** MongoDB is a document database that stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use.

  Effectively, we used this database to store our data so we can access it later for further analytics and to build our dashboard.

Figure II.17: MongoDB database logo

- **GStreamer:** GStreamer is a free and open source multimedia framework that is mostly used to develop media applications (streaming, non-linear editing, media playback, etc..). GStreamer is a framework that makes it simple to develop programs that handle audio, video, or both. It makes use of plugins to provide various codecs and other features. [**?**].

Figure II.18: GStreamer logo

- **Dash:** Dash is a productive Python framework for building web applications. Written on top of Flask, Plotly.js, and React.js, Dash is ideal for building data visualization apps with highly custom user interfaces in pure Python. It's particularly suited for our project since we work with data issued from Python code [**?**].

Figure II.19: Dash logo

II.2.4 Scrum Meetings Tools

- **Slack:** Slack is a proprietary business communication platform developed by American software company Slack Technologies. Slack offers many IRC-style features, including persistent chat rooms organized by topic, private groups, and direct messaging [**?**]. These different channels enabled us to share useful documents and papers, send updated codes and have quick calls.

Figure II.20: Slack logo

- **Trello:** Trello is a visual collaboration tool that provides the team with a shared view of any project. Through Trello boards, we created a shared space for our team to organize, collaborate, and share information to accomplish their business goals in a

flexible and fun way [**?**]. This tool enabled us to organize tasks and responsibilities to keep our projects on track: Interact with teammates on Trello boards, create tasks and ideas on cards, organize cards into lists (To do, Doing, Done, Meetings, Important,...)

Figure II.21: Trello logo

- **Google Meet:** Google Meet is an application where we hosted video meetings. This tool enabled us to have large meetings when we needed to have a call with all DNA employees and directors. It also enabled us to make live demos when we had to try our algorithm results with the laptop camera. Besides, it is easily accessible on any device. We had only to share an invitation link and join the conversation [**?**].

Figure II.22: Google Meet logo

Conclusion

This chapter is dedicated to introduce the hosting company "DNA Global Analytics". This provided a better idea of the nature of activities and services there in general, and the project context in particular. It is also an introductory part in which we presented the general tools through out the project main steps. The need of the tools have come up along the project realization. The upcoming chapter's aim will be to allow further understanding of the evolution of the state-of-the-art.

III

Evolution of the state-of-the-art

Introduction

Object detection is an important computer vision task used to detect instances of specific types of visual objects (such as humans, animals, or cars) in digital images. The purpose of object detection is to develop computational models and techniques that can provide one of the most basic information required by computer vision applications. Other tasks that are commonly linked to object detection are the re-identification and image recognition of the object.

Research in CV focusing on either the object detection alone or on both the object detection and re-identification is growing rapidly thanks to the technology advancement in AI in recent years. In this chapter we will learn about the evolution of the state-of-the-art used in Object detection, Person re-identification and image recognition starting from R-CNN and the yolo family to fairmot detection and re-identification and finally CNN and VGG for the image recognition.

III.1 Detection and Re-Identification

III.1.1 FairMOT

Fairmot is the recent breakthrough in multi-task learning applied to object detection and re-identification. this approach significantly outperforms by a long shot the the previous methods thanks to the technique Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, Wenyu Liu used in the creation of the this approach, which is is multi-task learning(MTL), a popular approach in ml literature that goes back to 1998 paper by Rich Caruana by the same name, So instead of doing the usual “detect first, re-ID secondary” both of those tasks are handled in parallel at the same time. In this section, we will dive in the technical details of FairMOT including the backbone network, the object detection branch, the re-ID branch as well as inference details.

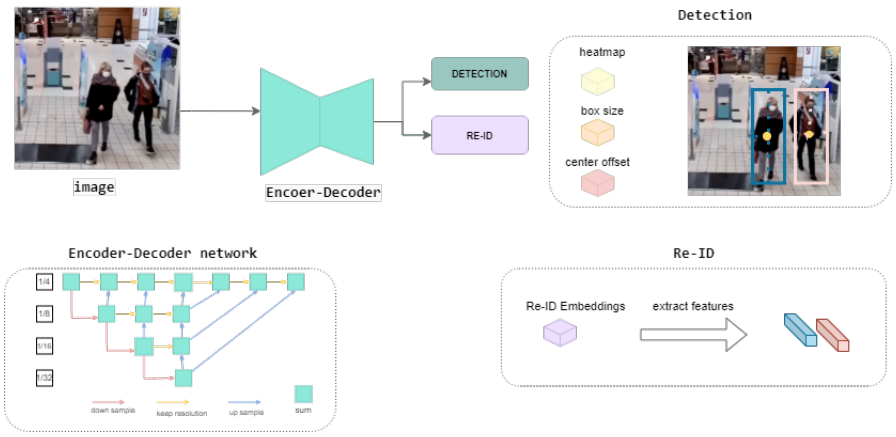


Figure III.1: The architecture of FairMOT

III.1.1.1 Backbone Network

ResNet-34 was picked as a backbone so that FairMOT has a good balance between speed and accuracy. An enhanced version of Deep Layer Aggregation (DLA) [10] is applied to the backbone to fuse multi-layer features as shown in the figure below Different from original DLA [20], it has more short cuts between low-level and high-level features which is similar to the Feature Pyramid Network (FPN) [45]. In addition, convolution layers in all up-sampling modules are replaced by deformable convolution such that they can dynamically adjust the receptive field according to object scales and poses.These changes are also beneficial in resolving the alignment issue. DLA-34 is the name given to the resulting model.

Assign the size of input image as Himage × Wimage, then the output feature map has the shape of C × H × W where H = Himage/4 and W = Wimage/4 détection branch

III.1.1.2 Detection branch

The detection branch is built over CenterNet [10] .We'll give you a quick rundown of how we went about creating this piece. self-contained. Three parallel heads are added in particular.

to DLA-34 for the purpose of calculating heatmaps, object center offsets, and sizes of the bounding boxes, respectively Each head is put into action. By applying 3 × 3 convolution (with 256 channels) to the output features of DLA-34, followed by a 1×1 convolutional layer which generates the final targets. Heatmap Head The position of the object centers is estimated using this head. The de facto standard for the landmark point estimate job is the

heatmap-based form, which is used here. The heatmap's size is 1 H x 1 W. If a heatmap location collapses with the ground-truth object center, the answer is anticipated to be one. As the distance between the heatmap position and the object center increases, the reaction decays rapidly.

For each GT box $\mathsf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we compute the object center (c_x^i, c_y^i) as $c_x^i = \frac{x_1^i + x_2^i}{2}$ and $c_y^i = \frac{y_1^i + y_2^i}{2}$, respectively. Then its location on the feature map is obtained by dividing the stride $(\tilde{c}_x^i, \tilde{c}_y^i,) = (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$. Then the heatmap response at the location (x, y) is computed as:

$$M_{xy} = \sum_{i=1}^{N} exp^{-\frac{(x-\tilde{c}_x^i)^2 + (x-\tilde{c}_y^i)^2}{2\sigma_c^2}}$$

where N represents the number of objects in the image and σ_c represents the standard deviation. The loss function is defined as pixel-wise logistic regression with focal loss [49]:

$$L_{heat} = \frac{-1}{N} \sum_{xy} \begin{cases} (1 - \hat{M}_{xy})^{\alpha} \, log(\hat{M}_{xy}), & if \; M_{xy} = 1 \\ (1 - M_{xy})^{\beta} (\hat{M}_{xy})^{\alpha} \, \mathsf{log}(1 - \hat{M}_{xy}) & otherwise \end{cases}$$

where $\hat{M}$ is the estimated heatmap, and α, β are the predetermined parameters in focal loss. Box Offset and Size Heads The box offset head tries to more precisely locate items. Because the final feature map's stride is four pixels, it will include quantization errors of up to four pixels. This branch estimates an offset relative to the object center for each pixel order to reduce the impact of down-sampling. The box size head is in charge of estimating the target box's height and width at each point.

Denote the output of the size and offset heads as $\hat{S} \in R^{W \times H \times 2}$ and $\hat{O} \in R^{W \times H \times 2}$, respectively. For each GT box $\mathsf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we compute its size as $s_i = (x_2^i - x_1^i, y_2^i - y_1^i)$. Similarly, the GT offset is computed as $o^i = (\frac{c_x^i}{4}, \frac{c_y^i}{4}) - (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$. Denote the estimated size and offset at the corresponding location as $\hat{s}^i$ and $\hat{o}^i$, respectively. Then we enforce l1 losses for the two heads:

$$L_{\mathsf{box}} = \sum_{i=1}^{N} \left\| o^i - \hat{o}^i \right\|_1 + \left\| s^i - \hat{s}^i \right\|_1$$

III.1.1.3 Re-ID Branch

The goal of the Re-ID branch is to provide characteristics that may be used to identify objects. Affinity between distinct items should, in theory, be less than that between the

same objects. To do this, we use a 128-kernel convolution layer on top of backbone features to extract re-ID features for each location. $E \in \mathbb{R}^{128 \times H \times W}$ is the the resultant feature map. The feature map may be used to extract the re-ID feature $E_{x,y} \in \mathbb{R}^{128}$ of an object centered at (x, y).

the re-ID features are learned through a classification task. The training set treats all object instances with the same identity as belonging to the same class. For each GT box $\mathsf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ in the image, we obtain the object center on the heatmap $(\tilde{c}_x^i, \tilde{c}_y^i)$. We extract the re-ID feature vector $E_{\tilde{c}_x^i, \tilde{c}_y^i}$ and learn to map it to a class distribution vector $\mathsf{P} = \{\mathsf{p}(k), k \in [1, K]\}$. Denote the one-hot representation of the GT class label as $\mathbf{L}^i(k)$. Then we compute the re-ID loss as: $L_{\mathsf{identity}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{L}^i(k)\mathsf{log}(\mathsf{p}(k))$, (3) where K is the number of classes. During the training process of our network, only the identity embedding vectors located at object centers are used for training, since we can obtain object centers from the objectness heatmap in testing

III.1.1.4 Network Inference

The network takes a frame of size 1088 × 608 as input and it predicts the object center using the heatmap head component and outputs the object box size (height and width) and the center offset. If the heatmap predicts one center the offset center will basically say shift your center a bit to the left or right and so on , this has shown to improve the accuracy of the detection branch due to eliminating anchors and using high resolution feature maps. The encoder-decoder outputs a compressed representation of the data passed in and then the network in a one fell swoop computes for every detected object the heatmap, box size,center offset and the re-id embeddings.

A re-id embeddings is extracted which is called a feature embedding these vectors are then stored in let us say a database and for every new detection it computes the nearest neighbor to all the embeddings in the database and that's how the re-id branch allows the system if an object disappears in the frame and afterwards reappears in it to detect it as the same object.

we introduced a framework to fairly balance the detection and re-ID that are done in one-shot without the need to separate the object detection and re-id task.Now we will proceed

with the more familiar option which is the first detect the object(using a deep convolutional network like R-CNN family and yolo) and assign the Id by passing it through a re-id component.

III.1.2 YOLO

Another approach to object detection is YOLO (You only look once). What the objects are in an image and where they are present can be predicted by looking only once at the image.

Instead of considering the task of detecting an object as that of a classification one, YOLO considers it a regression one in order to dimensionally separate the bounding boxes and associate their class probabilities [**?**].

A single network splits the image into many portions, generates bounding boxes and class probabilities for every portion being an object. It is capable of predicting bounding boxes along with their class probabilities from a picture in a single analysis.

Similarly, a single convolutional network is able to multiple bounding boxes and their class probabilities. Each bounding box is given weights based on the probabilities predicted. This network can be optimized from end to end based on the performance of detection because there is a single network involved [**?**] [**?**].

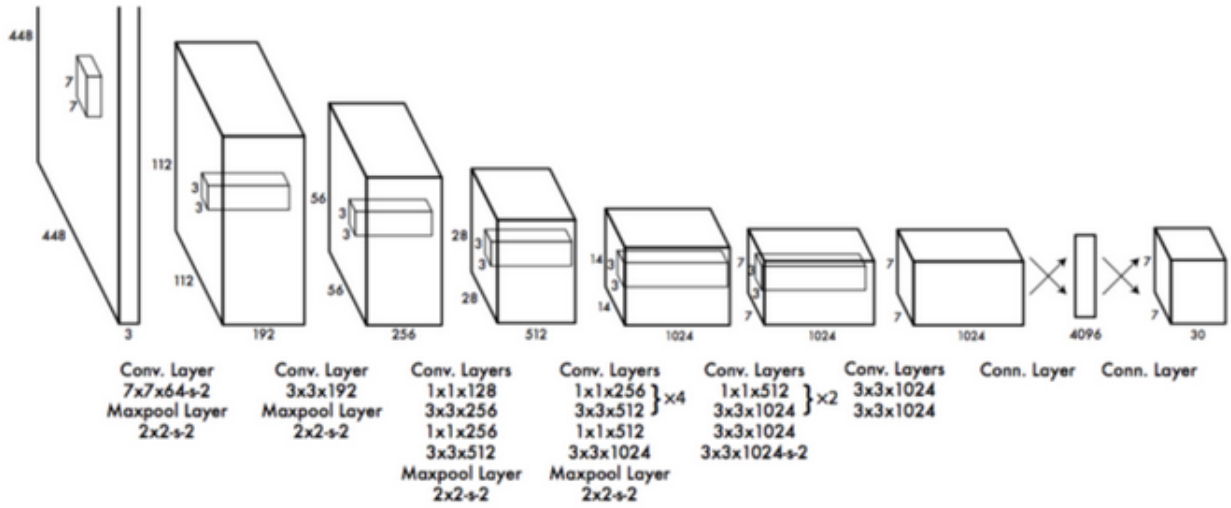


Figure III.2: YOLO Architecture

III.1.2.1 YOLOv1

The input image is divided into an SxS sized grid. If the center of a particular object falls in the center of a grid cell, then that grid cell has the responsibility of detecting that object [**?**]. The grid cell has the job of predicting bounding boxes(B), confidence scores and class probabilities(C) for each grid [**?**].

These predictions can be formulated as $\mathbf{S} \times \mathbf{S} \times (\mathbf{B} \times \mathbf{5} + \mathbf{C})$, which is a tensor. The confidence scores indicate the confidence level and accuracy level of the model in predicting that an object is contained in the box. The confidence score can be defined as Pr(Object) x IOUtruth pred where IOU stands for intersection over the union.
The confidence score is zero if there isn't any object that exists in that cell otherwise it should be the value of IoU between ground truth and predicted box [**?**].

During testing, conditional class probabilities and confidence predictions for individual boxes are multiplied in order to get class specific scores for every box. Pr(Classi|Object) x Pr(Object) x IoUtruthPred = Pr(Classi) x IoUtruthPred [**?**]. The network architecture of YOLO version 1 has taken inspiration from a network model named GoogleNet which was designed for image classification. This network consists of 24 convolutional layer sand 2 fully connected layers. The alternating 1x1 convolutional layers and reduction layers help

in reducing feature space from the preceding layers [**?**].

III.1.2.2 YOLOv2

YOLOv2 is a new method which is used to harness a huge quantity of classification data which is present in order to develop the scope of existing techniques. This technique allows the combining of distinct data-sets together using an ordered perspective of object classification.

A joint training algorithm enables the training of object detectors on detection as well as classification data. This technique allows the images to accurately comprehend the localization of objects and it makes use of these classified images to escalate its robustness and vocabulary [**?**].

In YOLOv2 firstly the classification network is fine-tuned on ImageNet at a full resolution of 448x448 for around 10 epochs. This helps the network in adjusting its filters so that it can work better on inputs with high resolution. The next step is the network fine-tuning. This classification network of high resolution gives a 4% increase in mAP. In this version of YOLO, the prediction of bounding boxes is done by anchor boxes instead of fully connected layers.

Firstly, a pooling layer is removed so as to escalate the network's output resolution. In Order to operate the network on 448x448 instead of 416 input images, the network is shrunk [**?**].

III.1.2.3 YOLOv3

YOLOv3 makes use of logistic regression to predict the objectness score for the bounding boxes. The objectness score should be 1 if a ground truth is overlapped by a bounding box prior greater than another bounding box.

The prediction can be disregarded if a bounding box prior overlaps the ground truth by a certain threshold but isn't the best. Boxes in YOLOv3 are predicted in three different

scales. Then the features are extracted from each scale in a technique similar to that of feature pyramid network. Feature maps belonging to the previous two layers are unsampled twice. The feature map which already exists within the network is concatenated with unsampled features.

This technique enables the gain of meaningful semantic and fine-grained information from unsampled features and previous feature map respectively. In YOLOv3, additional convolutional layers are included that process the combined feature map and to predict a twice sized tensor [**?**].

We will delve into this model's details in the next chapter since it has the best performances so far.

III.1.3 Image Classification and recognition

Convolutional networks (ConvNets) also known as convolution neural networks (CNN) have recently enjoyed a great success in large-scale image classification and recognition which has become possible due to the large public image repositories, such as ImageNet (Deng et al., 2009), and high-performance computing systems, such as GPUs or large-scale distributed clusters. In particular, an important role in the advance of deep visual recognition architectures has been played by the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) , which has served as a the ground for a few generations of large-scale image classification systems, from high-dimensional shallow feature encodings introduced by Xerox Research Centre Europe (XRCE) ,the winner of ILSVRC-2011 to deep ConvNets introduced by Alex Krizhevsky naming the CNN architecture after himself, Alexnet is the winner of ILSVRC-2012 improving the accuracy with about 10% over the previous year. ILSVRC have seen huge improvement over the years going from 26% ImageNet classification error in 2011 with the XRCE to 2.99% classification error in 2016 by Trimps-Soushen the Winner in ILSVRC 2016 The First to Obtain Error Rate Under 3% surpassing human-level performance (5.1%, Russakovsky et al)

III.1.3.1 VGG-16

Vgg is a convolutional neural network model submitted in the research paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman from the University of Oxford. The model achieves 92.7 % top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It's a bigger, deeper version of its predecessor Alexnet , they showed that smaller convolution with a larger number of them made a big difference. The improvements made over AlexNet are replacing large kernel-sized filters (11 in the first and second convolutional layer and 5 in the second convolutional layer) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks using NVIDIA Titan Black GPU's.

The architecture:

The input to the cov1 layer is a 224 by 224 RGB picture with a fixed size. The image is passed through a stack of convolutional (conv.) layers with an extremely narrow receptive field: 3x3 (the smallest size to capture the notions of left/right, up/down, and center). It also uses 11 convolution filters in one of the setups, which may be thought of as a linear transformation of the input channels (followed by non-linearity).The convolution stride is set to 1 pixel, and the spatial padding of convolution layer input is set to 1 pixel for 3x3 convolution layers so that the spatial resolution is retained after convolution. Five max-pooling layers, which are applied after part of the conv. layers, perform spatial pooling (not all the conv. layers are followed by max-pooling). With stride 2, max-pooling is done across a 2x2-pixel frame.

Following a stack of convolutional layers (of varying depth in different architectures), three Fully-Connected (FC) layers are added: the first two have 4096 channels each, while the third performs 1000-way ILSVRC classification and therefore has 1000 channels (one for each class). The soft-max layer is the last layer. In all networks, the fully connected layers are configured in the same way. The rectification (ReLU) non-linearity is present in all hidden layers. It should also be highlighted that, with the exception of one, none of the networks feature Local Response Normalization (LRN), which does not enhance performance on the ILSVRC dataset but increases memory usage and computation time.

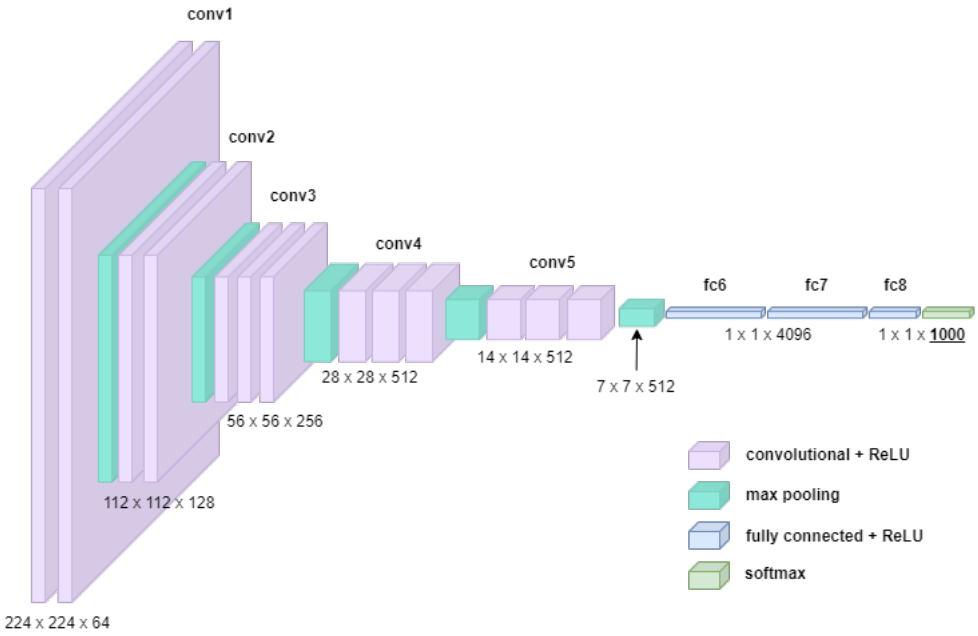


Figure III.3: the architecture of VGG-16

III.1.3.2 VGG-19

The VGG-19 Neural Network is defined as a type of neural network that is also specifically trained on over a million images from the ImageNet database, but the difference between the VGG-19 Neural Network and the VGG-16 Neural Network is that the VGG-19 Neural Network is 19 layers deep, whereas the VGG-16 Neural Network was 16 layers deep.

The above diagram is from the original research paper. The different columns from (A to E] show the different architectures researched by the authors . The column E refers to VGG-19 architecture. The VGG-19 was trained on the ImageNet challenge (ILSVRC) 1000-class classification task. The network takes a 3 channel 224 x 224 RGB image as the input, the same as for all VGGs.The difference between VGG-16 and VGG-19 is that:

- VGG-19 has 3 more convolutional layers split into the last 3 convolutional VGG blocks. The first added layer is in the 3rd convolutional block, a conv3 with an input 256x256. The second added layer is in the 4th convolutional block is also a conv3 but with an input shape of (512X512) the third and last layer is a conv3 with input shape (512X512).

- VGG-16 is already painfully slow to train and VGG-19 is slower .The network architecture weights themselves are quite large, furthermore vgg-19 has the highest weights of all the vgg family .
- The size of the VGG-16 network in terms of fully connected nodes is 533 MB, while the size of the VGG-19 network is 574 MB.

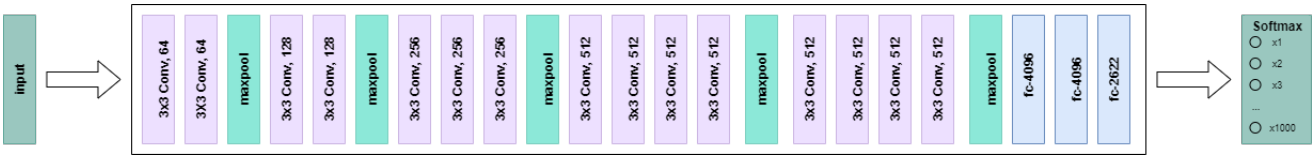


Figure III.4: The architecture of VGG-19

III.1.3.3 VGG-Face

The VGG-Face refers to a series of models developed for face recognition and demonstrated on benchmark computer vision datasets by members of the Visual Geometry Group (VGG) at the University of Oxford.The VGG-Face model, named later, was described by Omkar Parkhi in the 2015 paper titled "Deep Face Recognition." not to be confused with facebook's DeepFace which has fewer layers than VGG-Face.

The dataset that the researchers used contains 2622 identities of popular actors and for every identity they scraped the web for over 2000 pictures each using google and bing research engines totaling at over 2 million of images.

This dataset is then utilized to train deep CNNs for face recognition tasks including face identification and verification. Models are trained on the massive dataset and then tested on benchmark face recognition datasets, proving that the model is capable of extracting generalized characteristics from faces. They describe how to train a face classifier that classifies faces as humans using a softmax activation function in the output layer. This layer is subsequently eliminated, leaving a vector feature representation of the face, known as a face embedding, as the network's output. The structure of the VGG-Face model is demonstrated below. Only the output layer is different from the imagenet version with 2622 instead of 1000 .

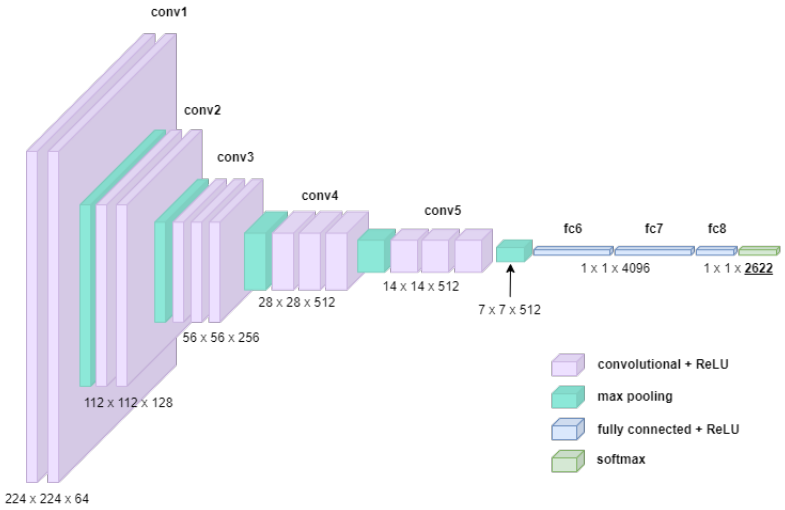


Figure III.5: The architecture of VGG-Face

After discussing the models above, I believe Using this VGG for our case which is estimating the age and gender of detected persons is the best since regular VGG is trained on imagenet dataset whereas VGG-Face trained on face dataset. In other words, VGG-Face comes with an advantage to recognize some facial patterns.

Conclusion

In this part of the report, we introduced the timeline of the evolution of the state-of-the-art models in the CV applications. First we begin with the object detection algorithms, starting by explaining FairMOT and its architecture which is one of the newest cutting-edge AI algorithms in multishot object detections and re-identification . Then we went through the more traditional way and introduced YOLO in it's 3 versions TK . second, TK we gave an overview of the VGG-Face model and explained briefly how in our case it can perform better than the previously mentioned algorithms.

At the end, and based on a comparison between the previous models we decided to use the once approach for object detection and re-identification which is FairMOT and use VGG-.face for the age and gender estimation. They will later be introduced in the next chapter for solving our CV tasks.

IV

Proposed Solution

Introduction

In this chapter, first of all we will present the system design of our solution architecture, secondly we will introduce the proposed solution for our CV project that can be divided into two major parts, counting the arriving and leaving customers and then estimating both their age and gender. The first part of our solution consists of two tasks:

- Person detection and re-identification followed by person tracking and direction classifier.
- Age and gender classification using CNN.

We will now introduce the system architecture diagram followed by the detailed proposed solution.

IV.1 System design

IV.1.1 system architecture

We aim for our system architecture design to be a clear description and representation for our project idea. We've spent a considerable amount of time with the head of Engineering and my supervisor to come up with a concept that is best suited for our client budget and strategy.

The figure below showcases the overall system, in short we will connect the store's CCTV cameras to edge devices capable of running complex AI algorithms in order to process the live CCTV footage of arriving customers as input to our previously built deep learning models. The output of our models will be transferred in real-time to a cloud database for storage and further processing to be later visualized in a dashboard available to the client.

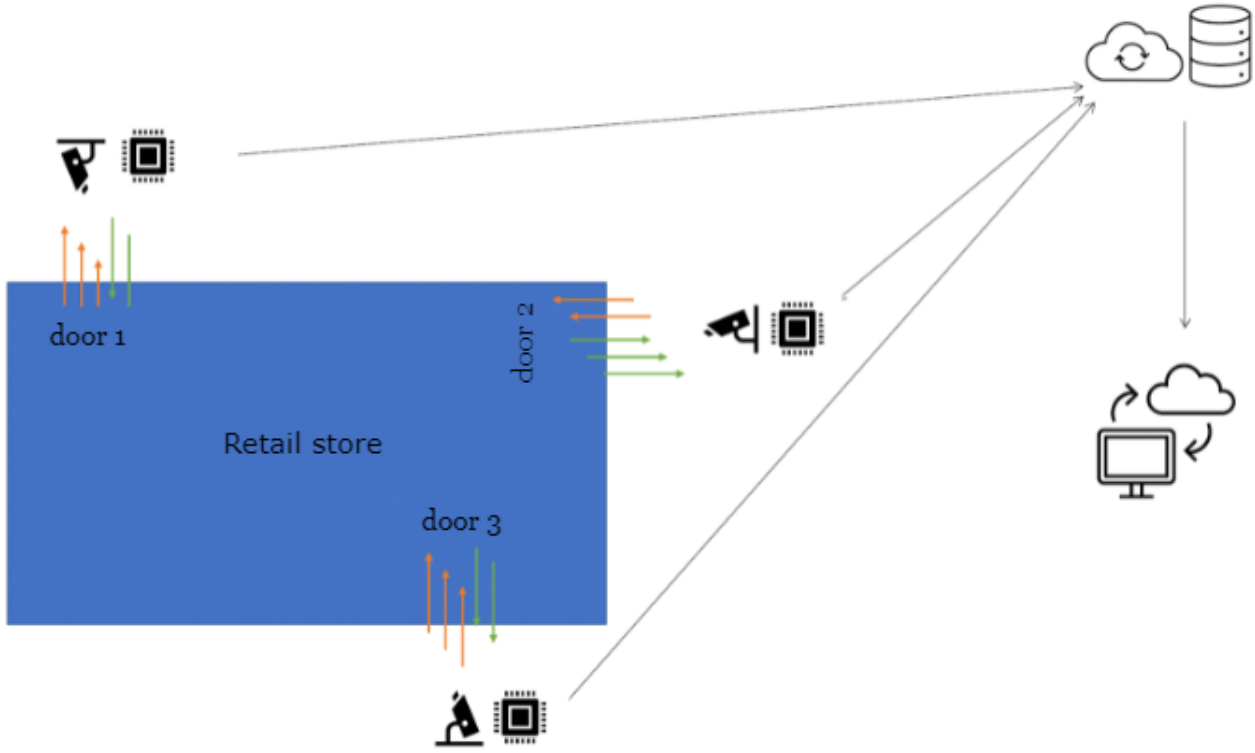


Figure IV.1: system architecture diagram

IV.1.2 system Value chain

Through the value chain we dig deeper into the different sub-systems developed to shape the overall system application by listing the technologies we've adopted in our design. Starting with the retail store cameras, the input of our system-application, we wanted to make use of the current camera installation of the Carrefour retail store for two main reasons, one being it's cost efficient for the retailer and the other being a business strategy to lure more client in since our system installation doesn't require to replace existing installations.
The next link in the chain is the edge device, where the input: camera video stream will be processed. The edge device will be responsible for 3 fundamental tasks, the first one is transmitting in real-time the CCTV live stream. The second task which is the core of our project is detecting, tracking and then counting the persons in the live, classifying then direction as entering or leaving the store while estimating their ages and genders. The third and last task is transmitting data outside the local network to a database where it will be stored in a cluster.

The last two links of the chain are the server that will request the data in the database cluster and update in real-time the graphs in the application dashboard.

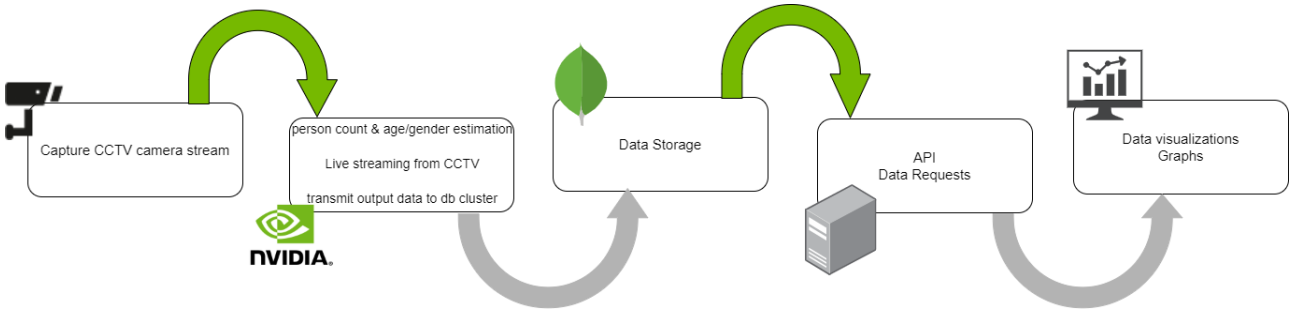


Figure IV.2: system value chain

In the next paragraphs we will focus on the second link of the value chain since it's the core of our data science project.We will detail the artificial neural networks applied to analyze visual imagery from the CCTV how they work and how we reimplemented them to achieve powerful AI models cable to detect,track,count and estimate both age and gender almost like a human being.

IV.2 Proposed solution

IV.2.1 FairMOT for person Detection and re-identification

IV.2.1.1 Overview

FairMOT is a simple baseline composed of two homogeneous branches proposed to predict the pixel level target score and ReID features. As indicated by its name FairMOT achieves the fairness between the two tasks Detection and Re-Identification while obtaining a higher level of real-time MOT performance.

We will use fairMOT pre-trained model available in the Github repository of the paper authors. It was trained on the MOT15 which is a dataset for multiple object tracking containing 11 different indoor and outdoor scenes of public places with pedestrians as the objects of interest, with different camera angles, camera motion and imaging conditions. First we will define its architecture in detail followed by the model inference for both parts

of our CV application.

IV.2.1.2 FairMOT Inference

The input image is resized according to the network frame size 1088 x 608 then fed into it to extract the heatmap, weight, offsets, and RE-ID features. We perform non-maximum suppression (NMS) based on the heatmap scores to extract the peak keypoints removing duplicate detections with lower confidence. The NMS is fairly important as it adds positively to the mean average precision of the detections. It is achieved by performing a 3×3 max-pooling on top of the heatmap scores. we later extract the weights, offsets associated with the heatmap scores that are larger than a threshold. Then, we compute the corresponding bounding boxes.

IV.2.1.3 From points to bounding boxes

At inference time, we first extract the peaks in the heatmap for each category independently. We detect all responses whose value is greater or equal to its 8-connected neighbors and keep the top 100 peaks. Let $\hat{P}_c$ be the set of n detected center points $\hat{P} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n}$ of class c. Each keypoint location is given by an integer coordinates (x_i, y_i). We use the keypoint values $\hat{Y}_{x_i y_i c}$ as a measure of its detection confidence, and produce a bounding box at location :

$$(\hat{x_i} + \delta\hat{x_i} - \hat{w_i}/2, \quad \hat{y_i} + \delta\hat{y_i} - \hat{h_i}/2$$

$$\hat{x_i} + \delta\hat{x_i} + \hat{w_i}/2, \quad \hat{y_i} + \delta\hat{y_i} + \hat{h_i}/2)$$

where $(\delta\hat{x_i}, \delta\hat{y_i}) = \hat{O}_{x_i, y_i}$ is the offset prediction and $(\delta\hat{w_i}, \delta\hat{h_i}) = \hat{S}_{x_i, y_i}$ is the size prediction.

Figure IV.3: from keypoint heatmap to bounding box

The identity embeddings are also extracted at those selected centers. The bounding box is converted to the form of [x, y, Aspect Ratio, Height], and based on this, the Kalman Filters are initialized.

IV.3 Kalman Filtering for persons tracking

Using the detection and re-id algorithm described in chapter 3, we can obtain the central positions of moving persons as detected. Due to the noise and limitation of the detection method, such a central position representation sometimes does not accurately reflect the location of a moving object. When multiple objects occlude, the positions need to be further delineated in order to track each object. Through the tracking approach described in this section, we hope to obtain an optimal estimation of the path of each object, hence, to achieve robust tracking of multiple moving objects.

IV.3.1 Kalman Filter

Kalman filter is an iterative mathematical process that uses a set of equations and consecutive data inputs to estimate the true value, position, velocity, of the object being measured when the measured values contain unpredicted or random error, uncertainty or variation. It is mainly used to reduce the effects of the uncertainties due to wrong measurements.

Figure IV.4: Kalman filter iterative process

The Kalman filter is a uni-modal, recursive estimator. Only the estimated state from the previous time step and current measurement is required to make a prediction for the current state.

The flowchart above shows the recursive process. From the beginning, the filter keeps track of the State variable that contains the current value of the measurements being tracked (e.g. position, velocity), while the Process variable contains the predictive error of those measurements. Each time step, the state transition matrix moves the state and process matrix based on the current position and velocity, estimating a new position/velocity as well as new covariance.

From there, the Kalman Gain is calculated, along with the observed data. The update process involves using the Kalman in conjunction with the previous estimate and new observed data to update the state variable towards a belief that's somewhere between the prediction and measurement. The process covariance is also updated based on the Kalman gain. These updates are then used for the next round of predictions.

IV.3.2 Person Tracking

For each detection we initialize The Kalman Filters with the mean set as its predicted location and variance as a pre-defined uncertainty measure. For each successive frame, using

the Kalman Filter covariance and the new mean locations are predicted. Additionally, we calculate the embedding distance between old tracks and new detections. This is fused with the predicted mean and covariance to form the final matching. During fusing, we get the gating threshold based on the chi-square distribution value.

Chi-square distribution is a gamma distribution that gives an indication of the goodness of fit or how confident the new predicted results are when juxtaposed with the expected results. This is popularly used in inferential statistics and hypothesis testing.
Since the measurement has 4 degrees of freedom (x, y, Aspect Ratio, Height) the chi-square value would be approximately 9.5, representing a deviation of 0.05 which is the standard threshold used. The values greater than this threshold are marked with infinity.
Using Linear Assignment, each track is updated with a detection frame's values and based on the Kalman Gain, the mean and covariance are also updated. The unassigned tracks and frames are taken and passed to the next association based on IOU distance, following which a similar procedure as the above is repeated. The unassigned tracks in this step are marked as Lost. Finally, we initialize the unmatched detections as new tracks and save the unmatched tracklets for 30 frames in case they reappear in the future.

IV.4 Persons counter

For this task of our CV project we need the person tracking that we discussed in the previous sections and the line equation that classifies the general direction entering or leaving the store, the schema below visualizes the general concept .

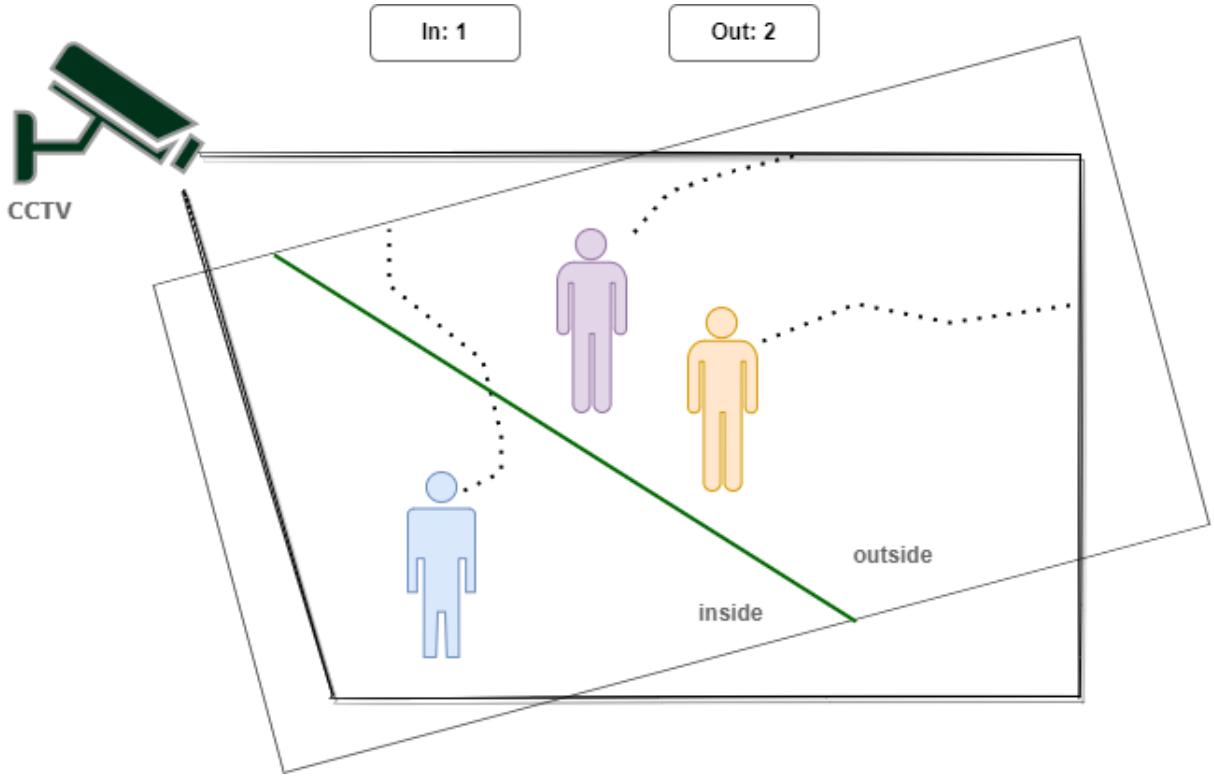


Figure IV.5: person counter illustration

in our case the CCTV camera is fixed so we went with a basic direction classifier we start with the y-coordinate value for all previous tracks for the person. Then we compute the direction by taking the difference between the current track and the mean of all previous tracks, then we interpret the polarity of the direction value as following:

- If the direction is negative indicating:
  - The person is going out the store
  - The person is going out the store
  - > in this case we increment OUT
- if the direction is positive indicating:
  - the person is going in the store
  - The track location is under the centerline.
  - > In this case we increment IN .

IV.5 Age and gender Estimation

Now we explore the second part of our CV application which is estimating the age and gender of the person. For this we will use transfer learning on VGGface model and imdb-wiki dataset.

IV.5.1 Transfer learning

Transfer learning has revolutionized the world of Computer Vision and made AI researchers migrate to this sub field of ML and thanks to it we can easily have amazing results with just fine tuning a small layer of a model using the pre-trained weights. In our CV project, we aim to classify the persons detected inside based on their age and gender. And since we have available by a click of a button hundreds of award winning neural networks trained on millions of images with state of the art performance we referred to some of these networks that best suit our project context as our pretrained weights in the Transfer Learning method, for example we can list VGG16, VGG19 and resnet-50.

Now for the dataset the award winning neural networks were trained on , we had few to choose from IMAGEnet, image database used in the ImageNet Large Scale Visual Recognition Challenge, COCO a large-scale object detection and segmentation and Vggface that contains thousands of face images.We opted for Artificial Neural Networks pre-trained on VGGface as they were trained on a dataset designed for facial recognition.
We transfer this intelligence from basic facial recognition to learning how to estimate age and gender. Hence, its name Transfer Learning. One of the most important strong points of transfer learning is that it gives us good results with far less data in a short period of time. Instead of training our model from scratch for thousands of epochs, we can use pre-trained weights and train our model for dozens of epochs to get better results.

Conclusion

Applications of Object Detection in domains like media, retail, manufacturing, robotics, etc need the models to be very fast so a compromise in accuracy to achieve fast computation is understandable.Now we theoretically paved the ground to experiment with all the models

we've researched retrain them and note the results, to later choose the best model achieving the best compromise between accuracy and speed.

V

Experiment and Results

Introduction

In this last chapter, we will present the general conception of the data science life-cycle that will be explained in detail in the next sections. After expressing the data science process, we will walk you through the first steps of our project beginning from data acquisition, model building and deploying it on our edge device, an NVIDIA Jetson AGX Xavier. Finally, we will show the results we obtained illustrated in the final dashboard.

V.1 Global Conception

As we are addressing a computer vision project, which is a field of artificial intelligence that trains computers to interpret and understand the visual world.And like any field of Artificial intelligence we need to follow a data science process to successfully build and deploy the AI models we need.

Figure V.1: Data science life-cycle

Let's define each of the data science process steps present in Figure V.1.

- **Obtain:** simply put, it is the very first step of a data science project. We obtain the data we need from available data sources. With the previously mentioned tools in Chapter I and the appropriate techniques, we are going to collect a huge data-set which will enable us to get good performances after models training.
- **Scrub:** After obtaining data, the second immediate step is scrubbing data. Since our data is issued from different sources, with different characteristics (sizes, resolutions,...), we should scrub it in a way to make it as clean as ready to be the appropriate input

to the model. This is processed by removing the outlier data, gathering the resulting downloaded data in their representative folders,...

- **Explore:** Once our data is ready to be used, and right before we jump into AI and Machine Learning, we will have to examine the data. Therefore, we will need to inspect the data and its properties, compute descriptive statistics to extract features and test significant variables, etc.

- **Model:** Before reaching this stage, we have to bear in mind that the scrubbing and exploring stage are equally crucial to building useful models. Therefore, most of the time was taken for data preparation. At this step, we did train models to perform classification of age and gender.

- **Interpret:** We are at the final and most important step of a data science project: interpreting models and data. The predictive power of a model consists on its ability to generalise. Which means the way we explain a model depends on its ability to generalise unseen future data.

  Interpreting data refers to the presentation of our data to a non-technical layman. We deliver the results to answer the business questions we asked when we first started the project, together with the actionable insights that we found during the data science process.

V.2 Google cloud platform

Training a ML model needs a lot of resources that aren't available in the office computer which is a generic PC with 8G of RAM and no GPU. Besides, the AI team is working on multiple projects at the same time. Therefore a VM where we setup all what we need is crucial to satisfy the problem requirements. After preparing a benchmark of the different available VMs and comparing their price quality ratios , we have chosen GCP as the most suitable in our case.

V.2.1 VM creation

To create and configure the VM , we made the following steps:

1. After logging in, we should browse to the compute engine section of GCP website and click on create new VM instance.

2. In the create new instance page, we fill out each of the fields as follows:

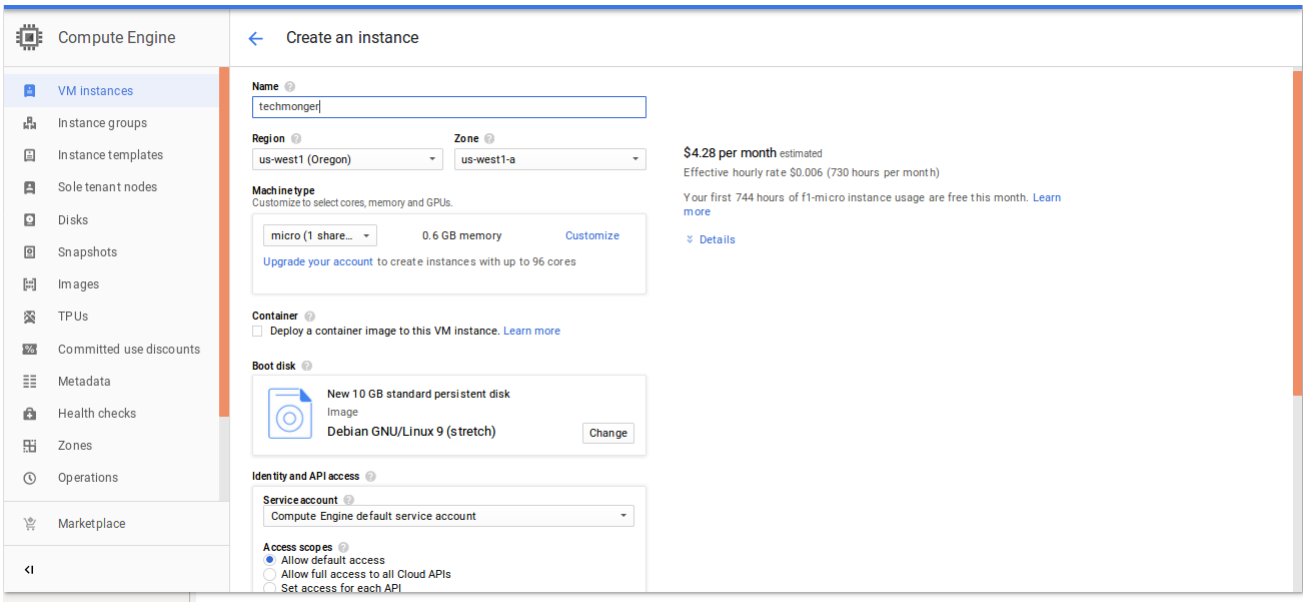


Figure V.2: google's compute engine for VM creation

(a) Name: DNA-GPU.

(b) Resource Group: DNA-GA in which instances share a common set of rules (e.g. network, security, etc.).

(c) Choose the operating system that will be installed on the VM we went with "Ubuntu Server 16.04 LTS".

(d) Region: Ouest Europe which is the nearest region having a GPU equipped VM.

(e) Machine type : Here we select N1 machine type with 4 vCPUs and 15 GB memory also including NVIDIA Tesla K80 GPU, RAM fixed at 24GB and temporary storage at 1TB. These settings are sufficient to run, build and train different models at the same time, with a cheaper price.

(f) We allowed http communication so that we can use jupyter notebook on our default browsers and we Defined the SSH public key so that all AI team members can authenticate connection to the VM.

(g) Public inbound ports where we allow the connection through the SSH port 22.

V.2.2 VM configuration

The previous steps allow us to successfully create the VM. The next task to do is to configure it to be ready for use. To do so, we have proceeded as follows:

1. We select the VM we have just created in the VMs menu.
2. We installed all the needed dependencies for the AI projects such as anaconda, python3 and jupyter notebook.
3. We connect through an SSH session and verify the installation has succeeded. **??**.
4. Upload the required configuration files, train code, dataset, etc to train the models.

V.3 Data collection

Our process to build our AI application has several steps. One of them is data preprocessing and analysis. We also need to select machine learning algorithms, train the model from scratch or train with transfer learning. However, often clients don't have data needed to solve their problem. More often than not, it is our job to get data from the web that is going to be utilized for model building.
Here we will discuss what we did in the data collection task, which falls into the obtain step from the data science process we've detailed earlier.

V.3.0.1 Dataset

Since we were not provided with custom client data we searched the web for public and open sources datasets to train our Age and gender classification models.

Dataset	total photo count	subjects	mixed ethnicity	gender label	age label
UTKFace	20.000	NA	yes	Binary	range (0 to 116)
Audience	26.580	2.284	yes	binary	8 age groups
Imdb-Wiki	524.230	100.000	yes	binary	range (0 to 110)

Table V.1: face recognition datasets

The Table V.1 presents the most used open source datasets for face recognition tasks. After comparing the datasets above we found that imdb-wiki is our best choice given its wide use in human face classification tasks , its total images count and its image quality.

IMDB WIKI open-source dataset is the largest publicly available dataset of human faces with gender, age, and name. It contains more than 500.000 images with all the meta information. All the images are in .jpg format. It was collected from imdb and wikipedia websites and holds thousands of celebrity pictures automatically labeled.

The problem with our dataset is that it's not ready for any machine learning algorithm. These are some problems that needs to be issued.

- All the images are in different resolutions.
- Some of the images are corrupted.
- Some images don't contain any faces.
- Some of the ages are invalid .
- The distribution between the genders is unequal (there are more males than females).

V.4 Data Pre-processing

Pre-Processing In machine learning focuses on the scrub step in the data science process,a step that is all-important to having a better dataset.
Every model has its own pre-processing methods that should be applied on the data before being fed to it. in order to pre-process our imdb-wiki dataset we went through various steps.

V.4.1 Data cleaning

Data cleaning is the process to remove incomplete data, incorrect data and inaccurate data from the dataset.

[6]:

	dob	photo_taken	full_path	gender	name	face_location	face_score	second_face_score
0	723671	2009	[17/10000217_1981-05-05_2009.jpg]	1.0	[Sami Jauhojärvi]	[[111.29109473290997, 111.29109473290997, 252....	4.300962	NaN
1	703186	1964	[48/10000548_1925-04-04_1964.jpg]	1.0	[Dettmar Cramer]	[[252.48330229530742, 126.68165114765371, 354....	2.645639	1.949248
2	711677	2008	[12/100012_1948-07-03_2008.jpg]	1.0	[Marc Okrand]	[[113.52, 169.83999999999997, 366.08, 422.4]]	4.329329	NaN
3	705061	1961	[65/10001965_1930-05-23_1961.jpg]	1.0	[Aleksandar Matanović]	[[1, 1, 634, 440]]	-inf	NaN
4	720044	2012	[16/10002116_1971-05-31_2012.jpg]	0.0	[Diana Damrau]	[[171.61031405173117, 75.57451239763239, 266.7...	3.408442	NaN

Figure V.3: loaded imdb-wiki data frame

The imdb-wiki dataset comes with a Matlab file".mat" containing all the meta information. The data frame shown in the figure above is generated through that ".mat" file.

- ➔ **dob:** date of birth (format: Matlab serial date number)
- ➔ **photo_taken:** year when the photo was taken
- ➔ **full_path:** path to image
- ➔ **gender:** 0 for female and 1 for male, NaN if unknown
- ➔ **name:** name of the celebrity in the image
- ➔ **face_location:** location of the face in the image.
- ➔ **face_score:** detector score (the higher the better). Inf implies that no face was found in the image and the face_location then just returns the entire image
- ➔ **second_face_score:** score of the second face detected in the image, will be used to ignore images with more than one face.

- **Handling Missing value:** for every NaN value in face_location, full_path or gender we simply omitted the row of data since due the very large amount of data we have, we can afford to lose a few rows here and there.

- **handling inaccurate data:** we removed pictures that do not include face, numerous faces, face scores and inaccurate gender binary numbers.

V.4.2 Data reduction

- **Numerosity Reduction:** Here the representation of the data is made smaller by reducing the volume without data loss. A way to numerosity Reduction in columns storing only numerical values is to change the data type according to the range of values. For example, in the case of our data, the minimum and maximum values of age are 1 and 100 respectively. This range of values can very well be represented by an 8-bit binary number. So, instead of storing age data as a 64-bit integer which is the default in most newer versions of Pandas ours is 1.1.5 , we can store it as an 8-bit integer. As the number of bits required to store the data has reduced, the memory usage also comes down. An 8-bit integer can range between -127 and +128 (in 2's complement representation), which will be sufficient for the age column in our dataframe. This will result in a significant reduction in the memory being taken up by the age column without losing any data information.
  This is also beneficial since this can reduce memory usage to a large extent, and can prevent the unnecessary occurrence of MemoryError in our program.

V.4.3 data transformation

- Age column needs to be generated from subtracting photo_taken from dob.

- crop faces from the images using the face location and resize them to (224x224) then finally saving them in pixels format under a new column named cropped_face_image_pixels.

- drop unused columns like dob, name , face_location,face_score,second_face_score, full_path.
  now we have a new data frame ready for the supervised training with:

  - **target:** gender for the gender model , age for the age model
  - **features:** a numpy array transformation of the cropped_face_image_pixels column.

- The final data is still unbalanced, male faces are more than female faces, this may generate in a model unable to generalize also biased to Males in the classification, so we decided to select similar number of male and female faces in the final dataframe.

- train test split is used to split the dataframe into train data used by the convolution neural network to learn and test data used to evaluate that model. We choose 80% train and 20% test with shuffle mode.

V.5 Model building

A Machine learning model is built by learning and generalizing from training data, then applying that acquired knowledge to new data it has never seen before to make predictions, in our case predict age and gender.
We opted for supervised learning. an approach to creating artificial intelligence, where a computer algorithm is trained on input data that has been labeled for a particular output, a target. The model is trained until it can detect the patterns and relationships between the input data and the target, enabling it to yield accurate predictions when presented with never seen before data.

V.5.1 Performance metrics

Performance metrics are a part of every machine learning and deep learning pipeline. They tell you how your model is doing and give you a number also known as a metric.
In every machine learning task, like selecting performance metrics, can be divided down into Classification or Regression . For both challenges, there are dozens of metrics to choose from, and it's up to us to choose the metrics that are best suited for our Model in order to understand how it interprets our data.

V.5.1.1 Classification metrics

Classification models have discrete output, so we need a metric that can compare discrete classes. Classification Metrics assess a model's performance and tell you if the classification is good or bad, but each one does it in a unique way.

1. **Accuracy**

   The simplest most familiar classification metric used is accuracy, which is defined as the number of correct predictions divided by the total number of predictions, multiplied by 100. It works very well only if the dataset is balanced meaning there are an equal number of samples belonging to each class.which is the case for us.

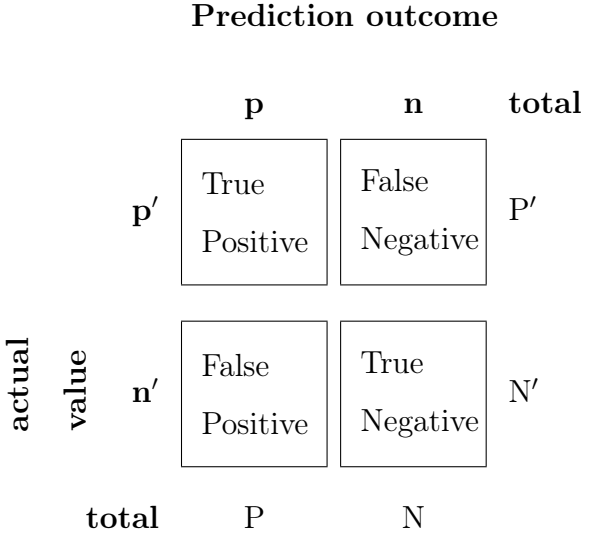
   Moreover, accuracy is the easiest to interpret by business stakeholders such as clients.

$$Accuracy = \frac{\text{Number of Correct predictions}}{\text{Total number of predictions made}}$$

2. **Confusion Matrix**

   Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

   Let's assume we have a binary classification problem. We have samples belonging to two classes like gender : female or male . After we run the classification model on our sample data we get the following results.



There are 4 important terms :

- **True Positive :** The cases in which we predicted female and the actual output was also female .
- **True Negative :** The cases in which we predicted male and the actual output was male .
- **False Positive :** The cases in which we predicted female and the actual output was male .
- **False Negative :** The cases in which we predicted male and the actual output was female .

3. **F1 Score**

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model.
Mathematically, it can be expressed as :

$$\text{F1 score } = 2 \times \frac{\text{Precision} \cdot \text{ Recall}}{\text{Precision} + \text{ Recall}}$$

- **Precision:** We can define the precision as the ratio of true positive (true predictions) (TP) and the total number of predicted positives(total positive predictions). The formula is given as such:

$$Precision = \frac{TP}{TP + FP} \tag{V.1}$$

- **Recall:** We can define the recall as the ratio of true positive(true predictions) and the total of ground truth positives. The formula is given as such:

$$Recall = \frac{TP}{TP + FN} \tag{V.2}$$

When True Positives and True Negatives are more relevant, Accuracy is employed, whereas F1-score is used when False Negatives and False Positives are more important.F1-score is used in unbalanced data.

V.5.2 Models Comparison

A comparative analysis is crucial to find out the model with best performances that fits our project. That's why we resumed the results in this table below:

Parameters	VGG-16	VGG-19	VGG-19(face)
Dataset pre trained on	Imagenet	Imagenet	VGGFace
Accuracy on train-set	89.67%	92.95%	95%
Accuracy on test-set	89%	92.41%	94.41%
Input Image Resolution	224x224	224x224	224x224
number of channels	3	3	3
Loss	0.2506	0.2304	0.1709
Time for prediction	1.5s	2.5s	-
prediction success for small-sized objects	Yes	Yes	yes
prediction success for large-sized objects	Yes	Yes	no

Table V.2: Models performances comparison

the vggFace dataset contains only images of faces in different angles on the other hand the Imagenet is a much standard dataset with over 1000 classes like variant animals species and objects of all sort. like the litterature suggested the model that was pre-trained on the VGGFace dataset had the best results, this is due to face features learned from the original

model beforehand. these specific already learned facial featured helped our trained model using transfer learning to evaluate and better classify age and gender.
This comparison allows us to choose VGG-19(face) as a detection model for our project since it outperforms VGG-19 and VGG-19 in terms of accuracy and loss. Add to that, its inference time is less than the other mentioned algorithms one which is a crucial point on our work since we need real-time prediction response.
Based on this analysis, we will detail the algorithm hyper-parameters used for our tasks.

V.5.3 Training hyper-parameters

A model hyper parameter is a configuration that is external to the model and whose value cannot be estimated from data often used in processes to help estimate model parameters. Since there isn't a rule to follow while setting these parameters, data scientists may use rules of thumb, copy values used on other problems, or search for the best value by trial and error. To boost training performances, we consulted various web blogs treating similar classification problems to find the best tweaks for hyper-parameters. We tried trying different values, the ones mentioned in the table below describe the best fit for our training model.

Parameters	Values
Initial learning rate	1e-4
End learning rate	1e-3
Total epochs	100
First stage epochs	50
Second stage epochs	50
Batch size	4
Image train size	416 x 416
IoU threshold value	0.5
Average decay	0.9
Gradient Optimizer	Adam
Train mode	GPU

Table V.3: Training hyperparameters

V.6 Deployment on edge device

We chose to deploy our project on an AI platform dedicated to autonomous machines and since our project requires high computational performance to compute. We decided to go with a gpu occupied board. Where else will we get the best gpus in the market but the GPU inventor itself, NVIDIA ?

NVIDIA Jetson ™ is a leading platform for stand-alone machines and embedded applications that unites high-performance, low-power Jetson modules, the NVIDIA JetPack ™ development kit for professional software acceleration and a complete ecosystem of sensors , SDKs, services and products for accelerating development. Compatible with the same AI software and cloud-based workflows used on other Nvidia platforms, the Jetson platform delivers the levels of performance and energy efficiency businesses need to implement defined stand-alone machines by software.
Nvidia offers a number of production-ready system-on-a-module (SoM) under the nvidia Jetson brand name. The jetson nano ,xavier and TX series are worldwide available and each jetson has aspects and use cases where it shines better.

V.6.1 Nvidia Jetson products comparison

The Jetson Nano is the first to be introduced having the specs detailed in the table below. It is the go to for students and researchers but it was not suited for production use due to its overheating issue and sloppy performance.
the jetson TX2 was later announced fixing the issues in the nano but the performances were still lacking. With its format as compact as the Jetson Nano, the NVIDIA Jetson Xavier NX module ,the latest addition to the jetson family, stands out as the worthy successor of the Jetson TX2. It benefits from new cloud-native support and accelerates the NVIDIA software stack in as little as 10 W with more than 10X the performance of its widely adopted predecessor, Jetson TX2. While introducing the Deep learning accelerator DLA that is used for offloading the inference effort from the module's GPU, a feature we are looking forward to using.

Finally the addition that have dethroned all previous jetsons , the Jetson AGX Xavier, Up to 32 tera-operations per second, a graphics card powered by 512 Tensor cores, 32 GB of RAM and 32 GB of storage capacity on eMMC 5.1 module... the Jetson NVIDIA AGX Xavier card easily outperforms the performance of the Jetson TX2 . While proving to be 10 times more energy efficient, a quality that will be a nice marketing touch to sell our project.

Since our product will be deployed in a device that will run continuously till the end of time and not an API on a server, the energy efficiency is very well needed along with computing that doesn't exhaust our processors ,consuming more power, causing it to heat up which leads to overheating.

	Jetson Nano Developer Kit	Jetson TX2 Developer Kit	Jetson Xavier NX Developer– Kit	Jetson AGX Xavier Developer Kit
AI Performance	0.5 TFLOPS (FP16)	1.3 TFLOPS (FP16)	6 TFLOPS (FP16) 21 TOPS (INT8)	5.5-11 TFLOPS (FP16) 20-32 TOPS (INT8)
GPU	128-core NVIDIA Maxwell™ GPU	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores	NVIDIA Volta architecture with 384 NVIDIA CUDA® cores and 48 Tensor cores	512-Core Volta GPU with Tensor Cores
DL Accelerator	NA	NA	2x NVDLA Engines	2x NVDLA Engines
CPU	Quad-core ARM A57 @ 1.43 GHz	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore	6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6 MB L2 + 4 MB L3	8-Core ARM v8.2 64-Bit CPU, 8 MB L2 + 4 MB L3
Memory	4 GB 64-bit LPDDR4 25.6 GB/s	8GB 128-bit LPDDR4 1866 MHz - 59.7 GB/s	8 GB 128-bit LPDDR4x @ 51.2GB/s	32 GB 256-Bit LPDDR4x | 137 GB/s
Power Consomption	5-10W	7.5-15W	10-15W	10-30W
Price	$99	$399	$399	$699

Figure V.4: Nvidia Jetson products comparison

V.6.2 Jetson Xavier AGX configuration

After purchase of the chosen Jetson Xavier AGX developer kit we proceeded with the configuration of the board and for that we needed a microSD card and a computer with built-in SD card slot or adapter and a stable internet connection. below are the steps followed to configure the jetson:

- **write system image to microSD**

  1. Download the Jetson Xavier NX Developer Kit SD Card Image from the nvidia developer official website.
  2. format the microSD card using SD Memory Card Formatter for Windows.
  3. Download, install, and launch Etcher to to write the Jetson SD Card Image to our microSD card like advised by jetson user guide.
  4. select " select image" and browse where the image was downloaded and click ok.
  5. insert the microSd and select "select drive" and choose the correct device.

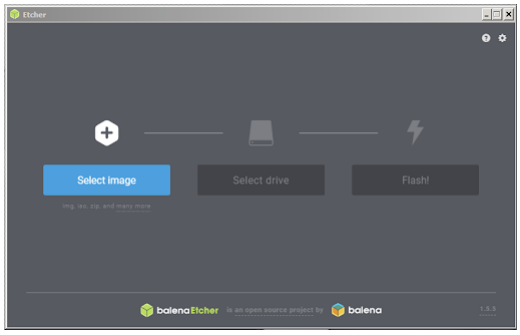


Figure V.5: write system image to microSD

  6. flash

- **setup the kit**

  1. Insert the microSD card we just prepared into the slot on the Jetson Xavier AGX module.
  2. Power the computer display and connect it.
  3. Connect the USB keyboard and mouse.

4. Connect the provided power supply. The Jetson Xavier NX Developer Kit will power on and boot automatically.

- **first boot**

  A green LED next to the Micro-USB connector will light as soon as the developer kit powers on. the Jetson Xavier NX Developer Kit will take you through some initial setup, when booted the first time including:

  1. Review and accept NVIDIA Jetson software EULA
  2. Select system keyboard layout, language, and time zone.
  3. Connect to Wireless network or use ethernet..
  4. Create username, password, and computer name.
  5. finally we Log in

After logging in we will see this screen, we are now ready to begin deploying our solution on the Jetson.

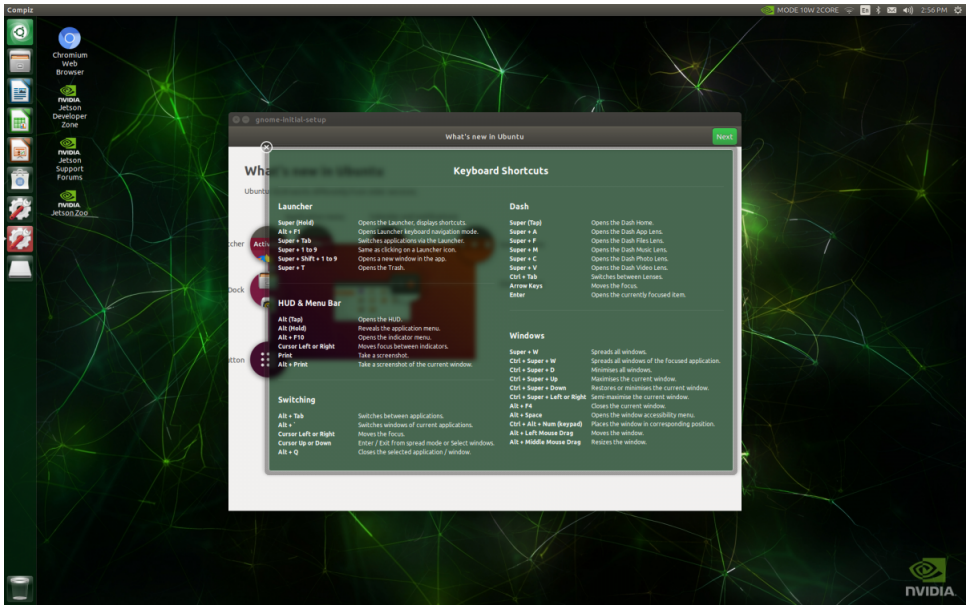


Figure V.6: Nvidia Jetson welcoming screen

V.6.3 Speeding up the deep learning inference

Data scientists excel at creating models that predict real-world data, but effectively deploying machine learning models is more of an art than science. In fact deployment

requires skills more commonly found in software engineering and DevOps. Venturebeat, leading source for transformative tech, reports that 87% of data science projects never make it to production. This highlights that deployment is a critical factor which makes the difference between success and failure.
In these next paragraphs we'll discuss the changes we've made in both age and gender models and the pre-trained FairMOT model to make them production ready, in other words to deploy these networks with high performance, high energy efficiency and high accuracy.

V.6.3.1 TensorRT overview

NVIDIA TensorRT is an SDK for high-performance deep learning inference built on CUDA. It provides a deep learning inference optimizer and a runtime for deep learning inference applications with low latency and high throughput.
What's outstandingly unique about TensorRT-based applications is that they perform up to 40X faster than CPU-only platforms during inference. We will be using the new TensorFlow-ONNX-TensorRT workflow to deploy our models. This workflow implies that the only thing you have to do is convert age and gender models to the ONNX format and use the ONNX parser in TensorRT to parse the model and build the TensorRT engine.Also, we did the same to our pre-trained FairMOT model converting it from a pytorch file extension to onnx file extension

V.6.3.2 ONNX overview

ONNX is a machine learning and deep learning model open format. It enables you to transform deep learning and machine learning models from a variety of frameworks, including TensorFlow, PyTorch, MATLAB, Caffe, and Keras, into a single format.
We can look at it as the bridge between our training framework (tensorflow) and the accelerator,TensorRT. Without this bridge, getting to the accelerator and creating the tensorRT engine will be a tedious task that will cause us tremendous headaches.
our workflow to create a tensorRT engine consists of the following steps:

1. Save both age and gender models as ".onnx"
2. Create a TensorRT engine(using nvidia developer available documentation we created a script that:

- Builds an empty network builder.create_network())
- The ONNX parser parses the ONNX file into the network parser.parse(model.read())
- set the input shape and then the builder creates the engine engine = builder.build_cuda_engi to create the TensorRT engine.

3. Run inference from the TensorRT engine(The TensorRT engine runs inference in the following workflow).

   - Allocate buffers for inputs and outputs in the GPU.we load the camera stream frames to buffers in the host using the load_images_to_buffer function.
   - Copy data from the host (memory attached to the CPU) to the allocated input buffers in the GPU using cuda.memcpy_htod_async(d_input_1, h_input_1, stream).
   - Run inference in the GPU using context.execute.
   - Copy results from the GPU to the host using cuda.memcpy_dtoh_async(h_output, d_output, stream)).

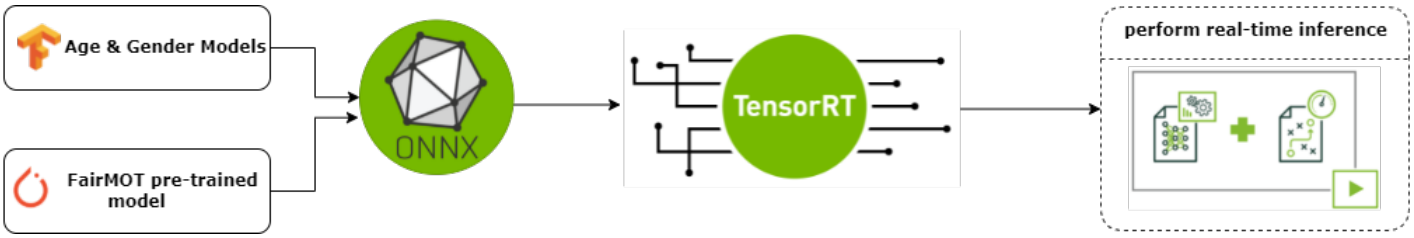


Figure V.7: Model inference process

V.6.4 Streaming the camera video footage

One of features our client wanted on the final project is the possibility to view the video stream from the CCTV camera that is being processed in the Jetson. For that reason we implemented a streaming component into our module so that while the camera stream is being processed it is also sent to a streaming server that will later on be requested to send that stream to our client (in our case into the final dashboard).

Managing multimedia is a difficult endeavor. As a result, choosing the right framework to process audio and video streams is critical to a project's success.

Before choosing the correct multimedia framework, we evaluated the following factors: coding complexity (architecture), cross-platform support, multimedia technologies coverage (codecs, filters, etc.), documentation, and support availability.

GStreamer excels in all of the above criteria: its intelligent plugin design and comprehensive core library make application creation simple, and it offers you with well-tested parts for many of your requirements. It runs on all major operating systems (e.g., Linux, Android, Windows, Mac OS X, iOS) and hardware architectures (e.g., x86, ARM, MIPS, SPARC). It comes with a large number of multimedia plugins (encoder, decoders,etc) and enables for easy integration of third-party plugins.

Finally, GStreamer has a large and well-structured documentation available for developers, as well as a large community. and these are the reasons why we chose gstreamer, the most used framework in embedded systems, for its versatility in our project.

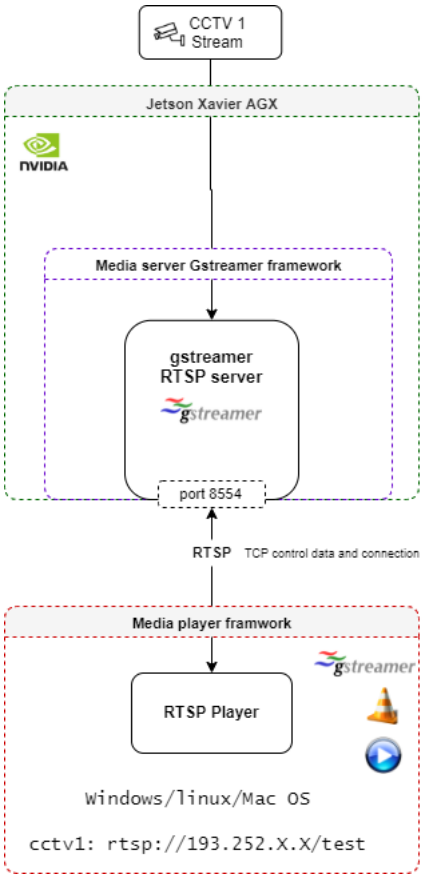


Figure V.8: RTSP client-server architecture using Gstreamer

- **steps to broadcast the streaming :**

  We wanted to be able to broadcast the stream from anywhere (internet and else)

outside our local network so we chose the port 8554 similar to the reserved 554 TCP used for accepting incoming RTSP client connections and for delivering data packets to clients that are streaming by using RTSPT.

1. Install GStreamer Python bindings are missing. Install them by either:sudo apt-get install python3-gst-1.0
2. install the gst rtsp server gir1.2-gst-rtsp-server-1.0
3. check if the ports are open : ss -tulw
4. initialize the camera by restarting it sudo systemctl restart nvargus-daemon
5. List report files opened by Internet connection and network on specific ports. here it lists the files opened by network connection or Internet using port 8554.

- **steps to receive the streaming:**
  After configuring the gst server and coding our python streamer the only step left is to receive it and since we designed a proper port we can use the internet to do so . From any PC with VLC, ffplay or gstreamer installed we can retrieve the flux like so.
  - VLC: vlc -v rtsp://193.252.X.X:8554/test
  - ffplay : ffplay rtsp-transport tcp rtsp://193.252.X.X:8554/test
  - gstreamer: gst-launch-1.0 playbin uri=rtsp://193.252.X.X:8554/test

V.6.5 Database

The whole purpose of our project is to collect data and build a customer's database but how can we achieve this goal if we don't implement a scalable, robust, always available database infrastructure ?

We decided on the non relational database MongoDB for so many reasons, first being a document-oriented database that stores data in json-like format unlike relational databases that store data a table format.

Second being of high scalability throughout horizontal scaling, a feature we will definitely need when the data fluctuates in size. The third reason is the platform and language support. Since we are coding in python 3 the database we ought to choose must provide an official fully supported driver and a large community for when we run into countless errors. Which

is thankfully the case for python mongo driver.
Finally, since the database is also shared with the IT team, the choice was agreed upon from both parties.

- **Mongodb Atlas:**
  MongoDB atlas is a globally available cloud database that lets you host your Database Clusters and connect to them remotely.It runs on the leading cloud providers like GCP, Azure, and AWS in which we can deploy our data through 80 regions worldwide.Going back to our project, we logged in to our atlas account, added a new project then we built a new cluster and selected AWS as our provider and went with the nearest region available where our customer's data will be stored.

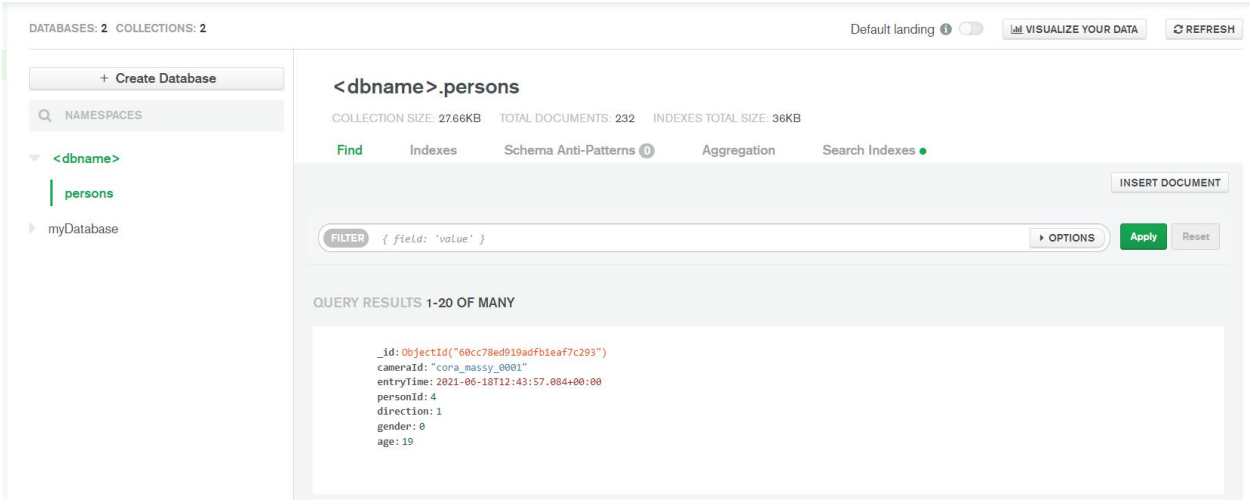


Figure V.9: Mongo db Atlas database view

Thereafter we set the connection method to connect our cluster, we need a URI that we will use later with the python driver to access the database and dump the data documents. So we choose " Connect your application" as our connection method and python as our driver and we copy the auto generated URI string that we will need in writing the script.

- **Pymongo:** Now that we created our cloud database then the next step is to go back to the Jetson Xavier where we will install PyMongo. PyMongo is a library that allows

interaction with the MongoDB database through Python; it is a native Python driver for MongoDB. These are the steps followed to configure the database connection.

- Install pymongo via pip3
- In a python script we imported the pymongo client and initialized it to access our cluster using MongoClient method and providing the URI we prepared earlier as the argument .

The script: because data is so important to us and because internet connections can easily be lost when problems occurs we wrote our script in a way that while the program is running on the jetson if the data failed to transmit to our cluster then it will be stored in a local json file "data.json" and when the connection is reestablished it will retransmit the document to the cluster. and that's one of the reasons mongoDB was our choice , handling json files is crucial in this project.

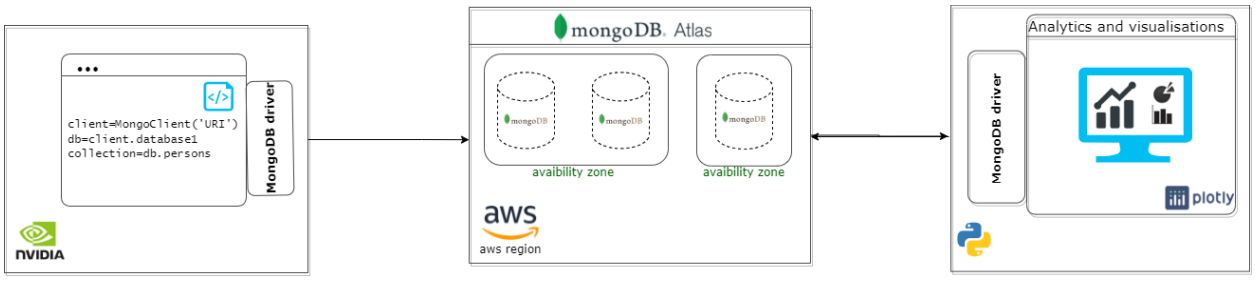


Figure V.10: Database architecture

V.6.6 Threading

the problem with computer programming is that it executes one line of code at a time and it doesn't move to next line until it executes the line above it, always doing one thing at once. Since our computer programming Language of choice is Python, an interpreted language that means the Python interpreter opens the file and starts reading it line by line, this can be major set back considering our program needs to do multiple things at a time to ensure a smooth workflow between all the complex functions we've created.
We have the deep learning models (FairMOT,age and gender) running full speed on the gpu.on the CPU the gstreamer main loop that is continuously looping through the CCTV

camera frames being preprocessed in GPU and a database manager that waits on the model's output to finally send it to MongoDB cluster.
If we write our python script to execute line by line then the processor is sitting idle waiting for data.Thereby for our problem threading is an appropriate solution to run multiple I/O-bound tasks simultaneously.
These are the steps followed in our main script with the added threads .

1. call the predictor() function : runs the TRT models on gpu and outputs new person's bounding boxes, direction(going in or out), age and gender.
2. initiate the MongoDB database client first then the database manager
3. call the database_manager() that will insure the new person's data is either inserted into the cloud cluster if the network connection was not lost or the data is dumped in a persistence_file under the json format if the connection was lost, the database_manager() will later try to retransmit it to the cluster.
4. initiate and start the database thread, that will consume the queued data to mongodb.
5. initiate and start the gstreamer mainloop which will stream the queued frames.

V.7 Dashboard

Developing the dashboard is the final step of our project. No project is redeemed complete without having something concrete to look at and extract valuable insights from. Our dashboard resumes our work and gives an overview of the results in a comprehensible and more attractive way. The fact of extracting insights from the model results that will be displayed based on statistics taking charts, diagrams,... forms will enable our client to have a great idea about his customers, classify his client-portfolio and satisfy his marketing requirements.

V.7.1 Specification requirements

In order to produce our final dashboard to be delivered to the client, we should specify the functional and non-functional needs.

- Functional requirements: The functionalities that should be presented in the dashboard are resumed in the following bullet-points:
  - Visualize the data flow in real-time.
  - Visualize the aggregation results.
  - Visualize all the required KPIs.
  - Visualize the summary of age,gender and people counting statistics periodically.
- Non-functional requirements: Our dashboard should answer to these criteria:
  - The dashboard should be attracting and its elements shouldn't be crowded.
  - All the information have to be clear and comprehensive.
  - The platform interfaces should be responsive and fast. It shouldn't annoy the user. Executing time should be as short as possible.

Title	Visualize dashboard.
Main Actor	Cora France marketing and sales Manager.
Goal	Provide Cora France marketing and sales Manager a dashboard containing an overview of the aggregated data related to the customers.
Pre-condition	The user accesses to the dashboard.
Principal scenario	1. The user accesses to the dashboard. 2. An overview of the aggregated data appears. 3. visualize the people count in real-time . 4. visualize the age and gender count in reat-time. 5. The user chooses to visualise the history of the data gathering for both age and gender separately. 6. The user chooses to view the real time video footage from the cctv.
Exception scenario	The system can't fetch the data, the use case ends.
Post-condition	The user has browsed the different dashboard slides.

Table V.4: Visualize Dashboard

The Table V.4 presents the scenario the user will go through when dealing with the dashboard.

Conclusion

Throughout this chapter, we introduced the method we used in order to collect and clean our final data-set. We also compared different models that led us to choose the best one in terms of performances and complexity. Then, we exposed the steps that we followed in our training process and the different parameters that we went through.After that we walked

you through the complex process of deploying the models in an end device, the jetson Xavier AGX, while taking full advantages of it's processing power and streaming the cameras output from the local network to the internet thought RTSP. Finally, we presented the dashboard that we developed with Dash plotly in order to extract insights from the models results.

Conclusion and Perspectives

This report summarizes the work conducted in a project that falls in the context of a graduation internship. The project is called "TK" , and it was conceived and developed from the ground-up at DNA global Analytics -Geneva.

We began by introducing our project context, including the hosting organism and the basic concepts. We then focused on specifying our requirements and specifications, which paved the way to a detailed design study. After that, we presented the process of building a deep learning model and then the intricate work of deploying these models in the Jetson Xavier AGX. Lastly, we showcased our achieved work through a well designed dashboard dedicated for the client. It is safe to say that we were able to tackle all the challenges encountered.

Our project aims big, it aims toward an autonomous store, where we Utilize a person re-identification model to track customers moving around the store. Collecting the data from different cameras not just one CCTV, the solution can track customers' paths through the sections of the mall generating a heatmap providing instant and in-time data for retail optimization tasks including pricing, staffing, marketing, store layout and much more.furthermore the futuristic solution will allow us to track customers in every corner of the store and re-identifies him thanks to our choice of FairMOT as our person detector, the base of our project.