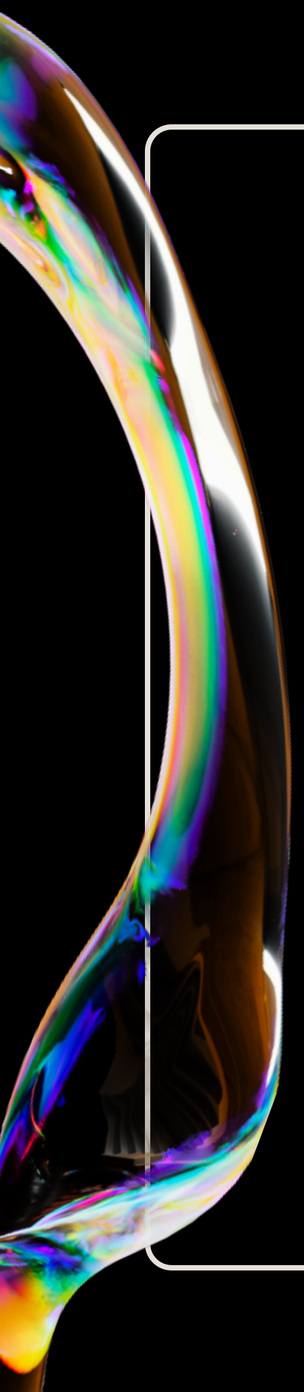
No name yet

yes its our team name

Abdalrhman Olimat
Ammar Saleh
Mohammad Salim



INTRODUCTION

We participated in the 42Amman AI Hackathon, where our goal was to build a face recognition system that identifies whether an uploaded photo matches a student from 42Amman based on their profile pictures fetched via the 42 API.

Main Objectives :

- Train a model to recognize 42Amman Students.
- Overcome challanges like limited data and image preporcessing.

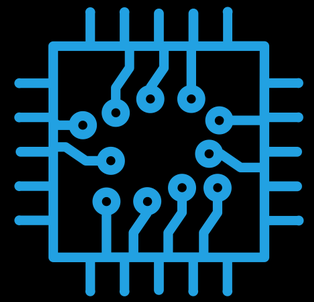
MAIN CHALLANGES

Limited training data

This was the Main challenge, with a very small dataset and with each student having just one picture, we dont really have enough data to capture variants and handle tricky situations.

Inconsistent image quality & overvitting risk

even though we reached a good accuracy with augmentation, but it was hard to generate good & accurate synthetic training samples to combat **overfitting and accuracy edge cases**.



Hardware challanges

although its considered a big challenge but we leveraged Kaggle Platform and it saved us.

WHAT WE DID

Model Development

- Built a face recognition model using Kaggle (GPU-enabled) and Python.
- Used FastAI's CNN_learner (ResNet34) for training.

Data Augmentation & Preprocessing

Since we had **very limited training data,** we used:

- PIL (Python Imaging Library) to:
  - Fix image orientations.
  - Normalize brightness/contrast.
  - Crop and align faces.
- FastAI's verify_images() to clean corrupted files.

Model Training

- Trained the model on 42Amman student images (from API) + 42Beirut images **(as negative samples).**
- Used FastAI's **cnn_learner** with ResNet34 (pretrained on ImageNet).

TECHNICAL IMPLEMENTATION

1. Data Collection & Preprocessing

Data Sources:

- **Primary Dataset:** Fetched 42Amman student profile pictures via the 42 API (OAuth authentication & Requests python library).
- **Negative Samples:** Added 42Beirut student images to improve model discrimination.

Preprocessing Steps:

- **Image Cleaning:** Used FastAI's verify_images() to remove corrupt files.
- **Face Alignment & Enhancement:** Applied PIL (Python Imaging Library) for:
- Cropping faces to a fixed aspect ratio.
- Adjusting brightness/contrast for consistency.
- **Data Augmentation:** Generated synthetic training samples using random flips, rotations, and slight distortions..

TECHNICAL IMPLEMENTATION

2. Model Architecture & Training

- Base Model: FastAI's cnn_learner with ResNet34 (pretrained on ImageNet).
  - Why ResNet34? Balances speed and accuracy, ideal for small-to-medium datasets.

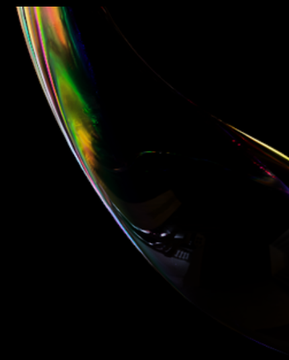
Training Process:

- Transfer Learning: Fine-tuned the pretrained model on our dataset.
- Progressive Resizing: Gradually increased image size to boost accuracy.
- Learning Rate Scheduling: Used lr_find() to optimize training efficiency...

TECHNICAL IMPLEMENTATION

3. Inference, Deployment and testing

- Deployed learn.predict() for real-time classification.
- Tested on unseen images (including side profiles and low-light conditions).
- Leveraged Kaggle feature to export the Model to be loaded offline and used anywhere



MODEL STRENGTHS

- ✅ Achieved good accuracy in distinguishing 42Amman vs. non-42Amman faces.
- ✅ Handled small dataset effectively with our augmentation techniques.
- ✅ Fast inference using FastAI's learn.predict().

MODEL LIMITATIONS

- ⚠ Dependency on image quality – Blurry or side-profile images reduced accuracy.
- ⚠ Limited to static images – No real-time camera support.
- ⚠ No Siamese Network (time constraints).

FUTURE WORK

- **Increase the Dataset Size**

A larger and more diverse dataset of 42 Amman students and non-students will help the model generalize better and reduce overfitting.
Including variations in lighting, angle, and facial expressions will make the model more robust in real-world scenarios.

- **Implement a Siamese Network for One-Shot Face Recognition**

Instead of retraining the model for every new student, a Siamese Network enables face verification by comparing the similarity between a new face and reference images in the dataset. This approach allows the system to scale efficiently, as it can identify or verify new students without needing to modify or retrain the entire model.

- **Integrate OpenCV for Real-Time Face Detection and Recognition**
  - OpenCV will allow:
    - Capturing live video from a webcam
    - Detecting faces in real time.
    - Cropping and preprocessing faces before passing them to the model.

THANK YOU

for your time and attention

No name yet team <3