ENID card detection and text lines segmentation

This solution consists of **three** essential models:

- **ID card segmentation model** (Instance segmentation model).
- **Card rotation (alignment) correction model** (classification model).
- **ID text lines detection model** (object detection model).

---

1. ID card segmentation model:

a. Dataset preprocessing (Dataset choice and reason):

- Final dataset used for training card segmentation model (data source):
  - Train: 6179
  - validation: 166
  - test: 74
- Challenges in data:
  1. Most available labelled datasets were designed for ID **detection** rather than **instance segmentation**. And the limitation of using the ID detection model is inability later to correct alignment of cards with high rotation angle (however, less than -/+ 90 degree).

Therefore, there was a potential solution for automatically annotating available data to fit an instance segmentation model and to overcome this limitation. by trying **SIFT** which is one of the commonly used algorithms for **object recognition** (template matching) and has the capability of detecting the objects with variant scaling and rotation angles.
However, unfortunately due to the bad quality of the data used during this experiment.
Finally, after further searching for data designed for instance segmentation of the ENIDs a good dataset was found and used for training the model (data source).

2. The second challenge is that all data points used for training the mode are in shades of Gray (but still 3 channels). However, surprisingly it was enough to train a model to be later adapted with coloured images.
3. Last limitation in the data, the labels are designed to detect only one class (0: ID). Therefore, the model will not have the capability to differentiate between the front and the back of the ENID. However, this limitation is managed later using the **Card Rotation Correction Model.**

b. Model implementation (choice) and training:

The final model used for ID card segmentation is the pretrained YOLOv11-nano model. Required 62 Epochs for training after applying early stopping with patience = 10.

c. Model Testing & Evaluation:

Test data: 74

Metric	Value
Mask mean IoU	0.935
Precision	0.986
Recall	0.973
AP@50	0.960

2. Card rotation (alignment) correction model:

To develop a robust ID card rotation correction system, the first step is to address 90° rotations by checking aspect ratios (width < height). However, this approach cannot detect 180° flips. To resolve this, training a **classification model** was required using a custom dataset of **correctly oriented** and **180° rotated IDs** (front/back) to ensure full orientation correction.

A classification model can classify 4 cases to correct 180 degree rotated cases:

0_correct_front, 1_180_front, 2_correct_back, 3_180_back.

a. Dataset preprocessing (Dataset choice and reason):

Dataset used for training this model was constructed in the following steps:

1. Collect **cropped** ID cards (front & back) from different data sources (from data used for ID segmentation model and lines segmentation model)
2. Create rotated cases, specifically 180 degree rotated ID front & back images.
3. Split data into train, validation and test set.

	Train	Validation	Test	Total
0_correct_front	25	10	5	40
1_180_front	25	10	5	40
2_correct_back	25	10	5	40
3_180_back	25	10	5	40
Total	100	40	20	

b. Model implementation (choice) and training:

To overcome the limited dataset size (160 samples), transfer learning was employed using the pretrained YOLOv11-nano classification model. The model achieved optimal performance at epoch 11 during training.

c. Model Testing & Evaluation:

Test set = 20 (5 data points per class)

	precision	**Recall**	**f1-score**	**Support**
0_correct_front	100%	100%	100%	5
1_180_front	100%	100%	100%	5
2_correct_back	100%	100%	100%	5
3_180_back	100%	100%	100%	5

3. ID text lines Detection model

a. Dataset preprocessing (Dataset choice and reason):

Data used for **front** ID lines detection model (data source) :

- Train: 589
- validation: 65
- Test: 35

Each label file contains 7 classes:

- 0: 'Code'
- 1: 'city'
- 2: 'family name'
- 3: 'name'
- 4: 'neighborhood'
- 5: 'number'
- 6: 'state'

Data used for **Back** ID lines detection model:

- Train: 50
- validation: 2

Each label file contains 7 classes:

- 0: 'back_id'
- 1: 'job1'
- 2: 'job2'
- 3: 'gender'
- 4: 'religion'
- 5: 'marital_status'
- 6: 'exp_date'

Due to the unavailability of pre-labelled datasets for text line detection on the back side of Egyptian National ID (ENID) cards, manual annotation was required. Using the VoTT (Visual Object Tagging Tool) software.

b. Model implementation (choice) and training:

A pretrained YOLOv11-nano detection model used in training of two separate models. The first model for detecting & extracting text lines of the **front** side of the ENID. And the second model for the **back** side. However, due to shortage of back side data, training required a higher number of epochs 100 epoch. while the front side model required only 50 epoch to reach the optimal performance.

c. Model Testing & Evaluation:

	Front	**Back**
mAP@50	0.9854	0.9950
mAP@50-95	0.7490	0.8149
Mean Precision	0.9832	0.9627
Mean Recall	0.9918	0.9988
Per-class mAP@50:	code: 0.9276	back_id: 0.9950
	city: 0.9950	job1: 0.9950
	family name: 0.9950	job2: 0.9950
	name: 0.9950	gender: 0.9950
	neighborhood: 0.9950	religion: 0.9950
	number: 0.9950	marital_status: 0.9950
	state: 0.9950	exp_date: 0.9950
Sample size:	35	2