Machine Learning
Project assignment

Mario Vento and Pasquale Foggia

The task

- For students taking the exam in January:
  - Car Make-Model verification
- For students taking the exam in February:
  - Age comparison

The teams

- Maximum 5 members per team
- All the members must take the exam in the same month

The deadlines

- For students taking the exam in January:
  - Project must be submitted by January 13
- For students taking the exam in February:
  - Project must be submitted by February 10

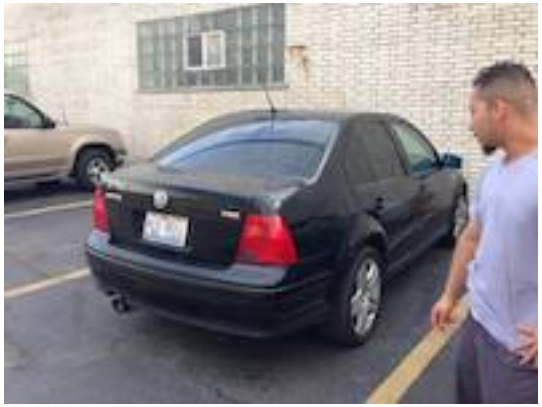
Car Make-Model Verification

- Given two car images, the developed system must check if they have the same make and model
  - Binary output (1=same make-model, 0=different make or different model)

- Difficulties
  - Cars with the same make-model may have different colors
  - Different lighting conditions
  - Car photographs taken from different positions (e.g. front, side…)

Car Make-Model Verification

Same make-model
Output=1



Different make-model
Output=0

Car Make-Model Verification

- Dataset
  - 40106 labeled pairs of images
    - 50% positive and 50% negative
  - 200x150 color images
  - MAKES: ['acura', 'audi', 'bmw', 'cadillac', 'chevrolet', 'chrysler', 'ford', 'honda', 'hyundai', 'infiniti', 'jeep', 'nissan', 'toyota', 'volkswagen']

Age Comparison

- Given the image of the faces of two persons A and B, decide if A is older than B, A is younger than B or they have (approximately) the same age
  - Three-class problem ("older", "same age", "younger"), the output is a 3-element vector with "one hot" encoding
- Difficulties
  - People may show the signs of their age in a very different way/measure
  - Different lighting conditions

Age Comparison

- Dataset
  - 45000 labeled pairs of images
    - Equally distributed among the three classes
  - 192x192 color images
  - Both male and female subjects

Age Comparison

Older than
Output=[0,0,1]

Same age as
Output=[0,1,0]

What you can use

- Anything (whatever kind of classifier, including non-neural ones; preprocessing, training strategy, validation,…)
  - But you must be able to explain what you have used…
  - It must be runnable inside Google Colab
- You can extend the training set with data augmentation and with new data
- You can use local computing power (you are not limited to Colab) for the training

What you must submit

- A link to a Google Drive folder containing:
  - The code used for training your system, in the form of a Google Colab Notebook
  - The files containing the saved models/weights after the training
  - The code needed to test the system (see next slide)
- You will also need to make a 8 minutes presentation of your system
  - It must be ready for the day of the exam, not for the project submission deadline

The verification script

- A Google Colab Notebook containing
  - The code to load your trained model
  - A "predict" function with the following specification:
    - Prototype: *predict(a, b)*
    - where: *a* and *b* are tensors containing the input images of the batch to predict; the shape of a and b is:

      (batch_size, height, width, 3)
    - return value: the tensor with the prediction on the batch; the shape of the return value is:

      (batch_size, 1) for car make-model verification

      (batch_size, 3) for age comparison

      [*batch_size* is the number of samples in the batch, and is not a fixed value; the function must work independently of this value]

How will the submissions be evaluated

- Using a test set disjoint from the training set
  - Not accessible to the students before the exam
- Computing the classification accuracy on the entire test_set
  - For car make-model verification, accuracy will be computed as: 1.0-abs(y_pred-y_true).mean()
  - For age comparison, accuracy will be computed as: 1.0-abs(y_pred-y_y_true).max(axis=1).mean()

Don't forget to

- Write the names of all team members in the Colab notebooks
- Ensure that the link you submit is readable to anyone (no authorization must be requested)
- Ensure that the predict function can take input images in the very same format used for the training set, and produces a return value in the format specified