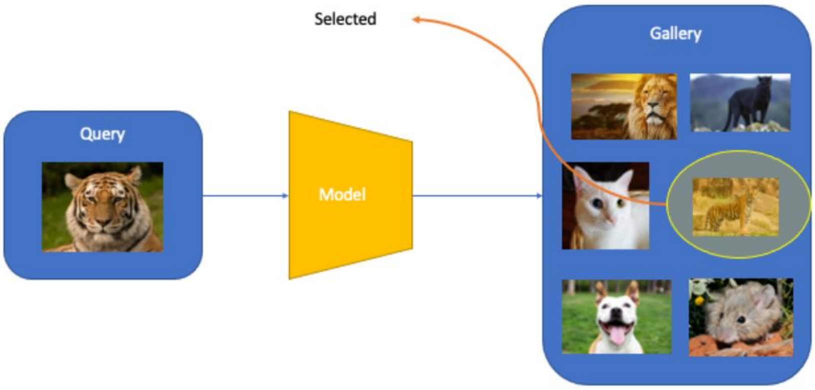
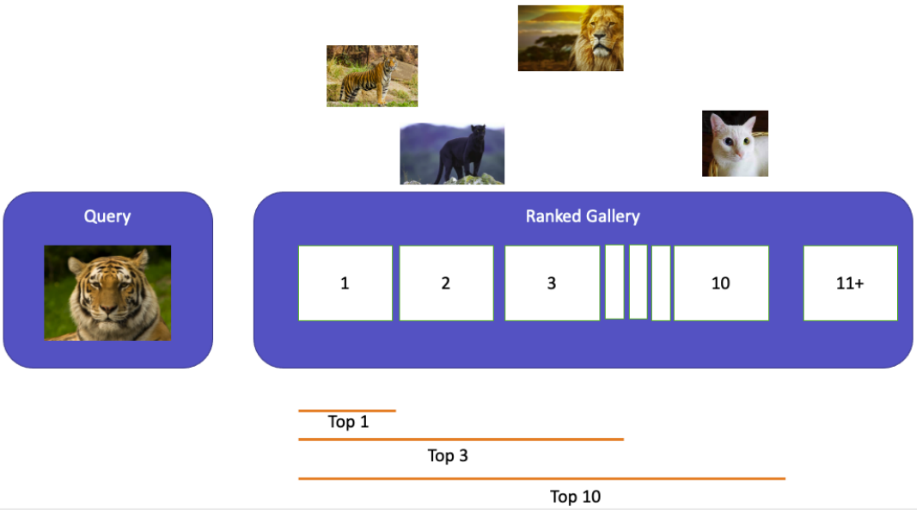
Introduction to Machine Learning Competition

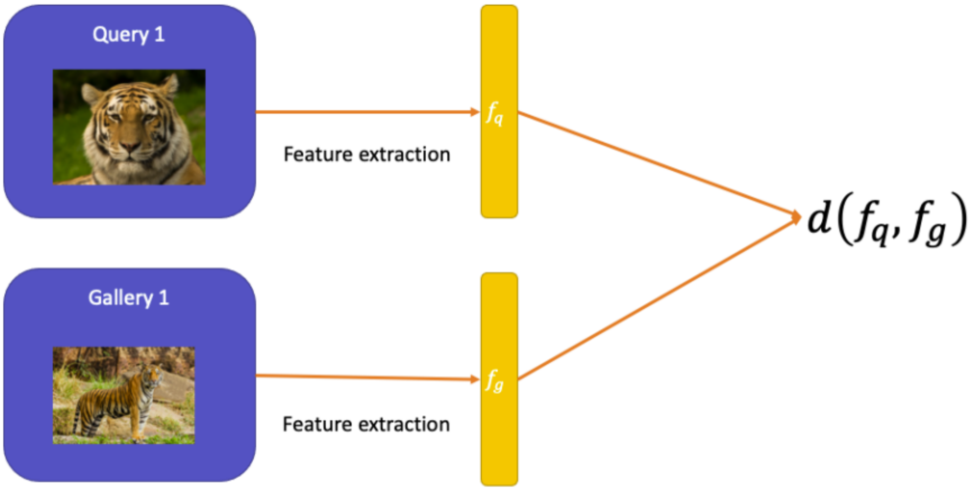
1. **Competition:** During the class we will release a brand new test dataset containing two folders: one for queries and one for gallery. Each group shall test their models and submit the results on a server that will reply with the achieved accuracy. The group that obtains the highest accuracy will win the competition. Here some useful information:
   a. The group shall show how the models have been tested, showing some **quantitative** and **qualitative** results.
   b. Each team shall submit a working code that shall produce similar to the presented results.
2. **Objectives:** the main objective of this project is to create an image search engine where a query image is fed to a model that will return the most N similar images from a gallery. The following example shows the problem of retrieving the same place based on a visual search algorithm.
   a. Given the input query image, the algorithm has to be capable of matching the input query image with another gallery image depicting the same animal.



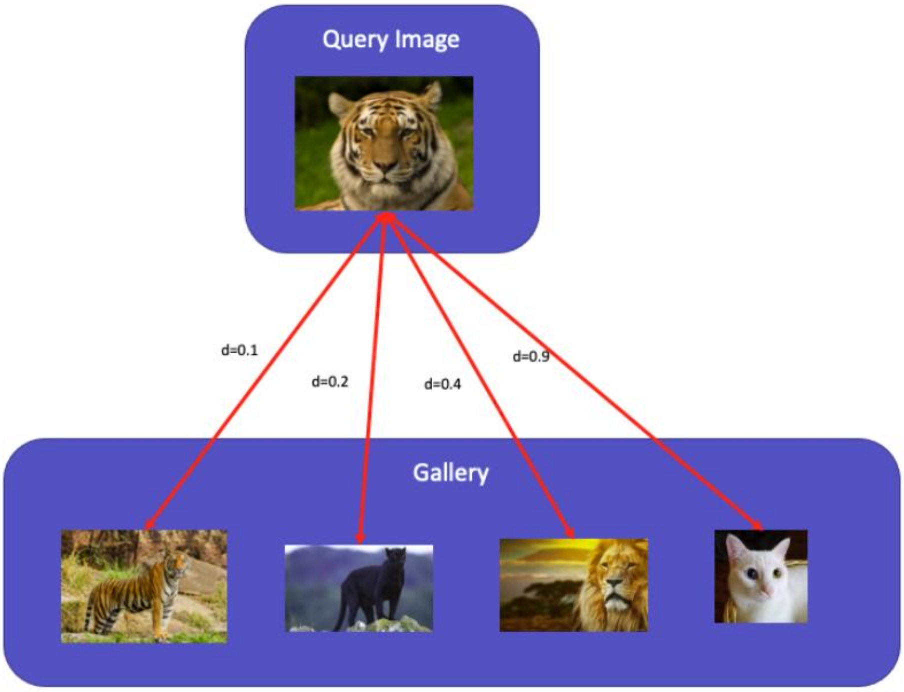
   b. The expected algorithm's output is therefore a list of ranked matches between the query image and the gallery images. An example output is depicted below where the top-k matches are reported. In this case, the algorithm correctly matched top-1 and top-2 images while the others (the ones reported) are false matches.



c. At this point it is important to define how a match can be defined. Among all the possibilities, one of the most used method is the definition of a similarity/distance d([], []) metric on top of extracted image features. As we saw during the colab example, once we extracted the query image features f_q, we can compute a q similarity/distance measure between our query f_q and each gallery f_g.



d. Once the feature distance between each gallery image Ig and our query image I_q has been computed, we can sort each match based on $d(f_q, f_g)$ matches as the top-k gallery images with the lowest feature distance from our query.



e. The given algorithm can be either pretrained on an external source of data (eg. a dataset of landmarks) or run simply at test time without any learned parameter. Please, note that the algorithm is not trained on the gallery data, otherwise this would be a simple classification ;)

f. A secondary objective of the project is to learn how to collect data. As we are seeing during the course, collecting a large amount of data is crucial for ML applications. Therefore, you are required to largely expand the provided training data in order to improve your algorithm performance on the test set. Let's see more details in the next section.

3. **Data:** we provide only an initial small amount of data but you are free to collect as much data as you like to train your model to improve your algorithm performance. The provided initial dataset is composed of a training set and a validation set. The training set is composed of 1300 images belonging to 10 different animals. The validation set has a gallery of 550 images with the 10 classes, a query of 100 images (10 for each of the 10 classes) and 200 distractor images. Finally, during the challenge day we will release the test set to measure the performance of your algorithm and to fill the leaderboard.
4. **Where to find initial data and example code:** the data and the code shown during the challenge presentation can be found in this zip. In the "dataset" folder you will find 2 sub-folders, one for each split. In the training folder, you will find a small amount of data that you are required to expand by adding data you will mine from external sources. In the validation folder you will find a gallery and a query. If you use Colab, you may need to have a look here.
5. **Methods to use:** you can use any of the methods you learned during the course (supervised/unsupervised, traditional/deep, etc). We encourage you not to limit the solution to just one method, but instead to try out different ones in order to better understand their strengths and limitations and report results obtained by each method.
6. **How to measure the performances:** the objective of the challenge is to obtain the best performance on the test split, namely, to match correctly the highest number of test query images in the test gallery images. To measure the performance of your algorithm we will use the top-1, top-3 and top-10 accuracy metric. The top-k accuracy metric allows to measure how often the correct match/label falls in the top-k matches/predictions and it's defined as:

$$Acc_{top-k} = \frac{C_k}{|Q_{val}|}$$

Where,

$$C_k = \sum_{q \in Q_{val}} 1_k(x_q, y_{top-k})$$

is the sum of correct matches among the top-k matches, Q is the test query set val and $1_k()$ is the correct match function between the top-k gallery matches y_{top-k} and a query sample x_q .

7. **How to measure the performances on the test set:** we will release the test set during the challenge day. Please note that you must not do the training of your model during the challenge day presentation. This may take long and you may not be able to complete your presentation in time.

For any further clarification please reach out to kumar.swaviman332@gmail.com.