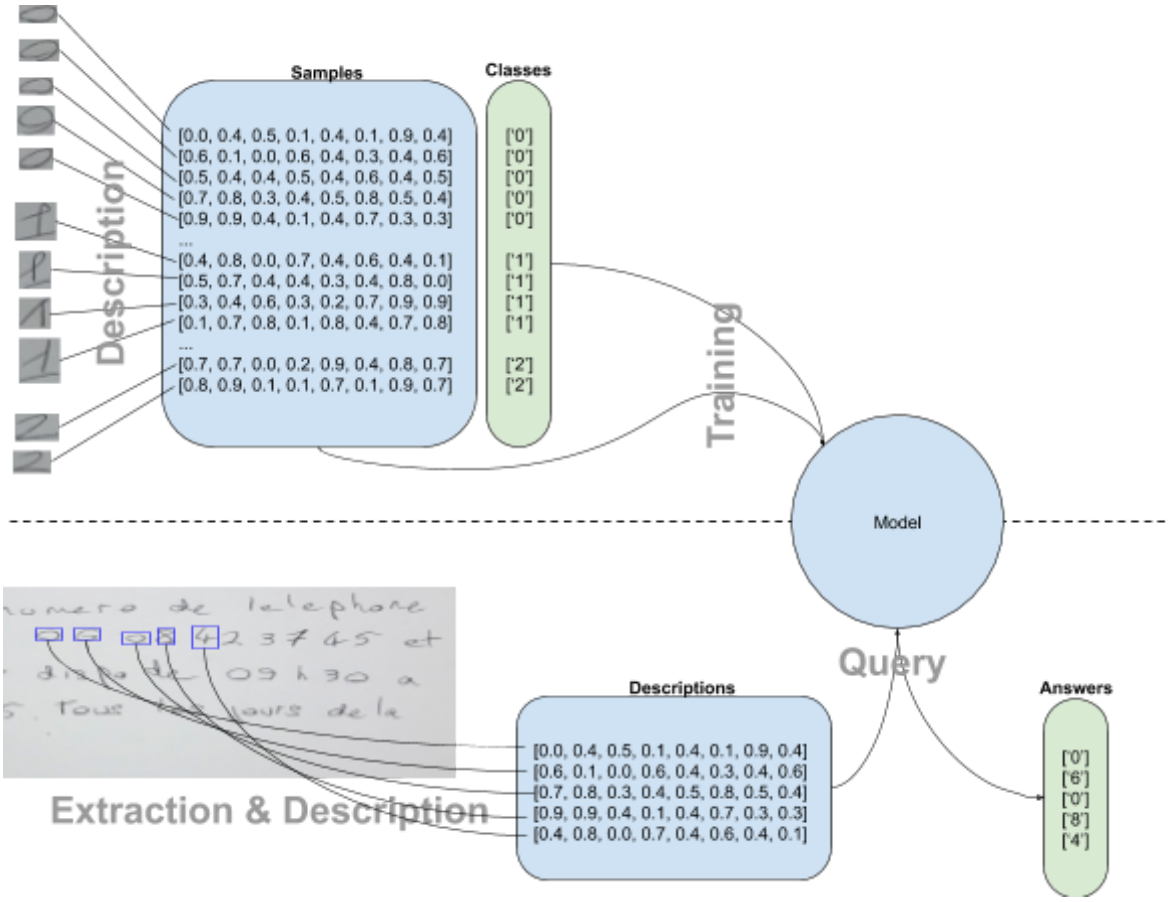
TP Vision par Ordinateur
Reconnaissance de caractères

Le but de ce TP est simple : détecter et reconnaître les chiffres manuscrits dans une image.

Pour cela, vous allez utiliser la détection et le matching de features, comme vu précédemment.
Vous pouvez télécharger le matériel de base pour ce TP à cette adresse :
www.benraynal.com/vision/ocr.zip
La technique consiste en 2 étapes :

- L'entraînement du modèle a partir d exemples
- La reconnaissance des chiffres grace au modele



Phase 1 : Entrainement

Pour entraîner un modèle, il faut lui fournir les descriptions (vecteurs de réels) d'exemples (ici, des images de chiffres) associés a leur classe (ici, le chiffre représenté par l'image).
Les données fournies dans ce tp comportent des répertoires nommés de 0 a 9, comportant des images des chiffres correspondants, ainsi qu'un répertoire X contenant des images de lettres quelconques. Le but de cette phase va être de

1. Charger les images de chaque répertoire
2. Calculer leur description
3. Entraîner un modèle en lui fournissant la matrice des descriptions, et la matrice des classes correspondantes

Etape 1a : Décrire un sample

Il existe plein de façons de décrire une image. Une des plus simple consiste à seuiller celle-ci, puis la redimensionner a une taille assez petite (pour réduire au maximum le nombre de pixels) mais pas trop (pour pouvoir différencier les différents chiffres). On prendra alors les valeurs des pixels ligne par ligne, comme descripteur.

Vous pouvez également y ajouter d'autres informations, comme le ratio hauteur/largeur par exemple.

Etape 1b : Entraîner un modèle

Une fois que l'on sait décrire une image, il suffit de creer une matrice des descriptions' ou chaque ligne contient la description d'une image, ainsi que la matrice des classes correspondantes (des nombres de 0 a 10, 10 correspondant au répertoire X).
Par exemple si chaque répertoire contient 10 images, les deux matrices auront 110 lignes.
Les 10 premières lignes de la matrice de description seront les descriptions des 10 images du repertoire 0, les 10 suivantes, celles du répertoire 1, etc…
Les 10 premières lignes de la matrice des classes contiendront un 0, les 10 suivantes un 1, etc...

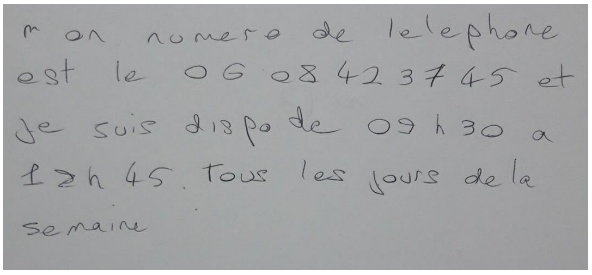
Il suffit ensuite de créer le modèle de votre choix, en lui fournissant les 2 matrices.
Vous pouvez par exemple choisir un modèle utilisant la **méthode des plus proches voisins** ,
un classifieur bayesien ou encore une **SVM**

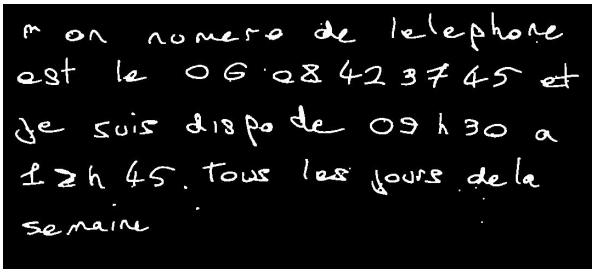
Phase 2 : Utiliser le modèle
Une fois que notre modèle est entraîné, on peut l'utiliser pour détecter des chiffres dans une image. Pour cela, il suffit de découper la sous partie de l'image correspondant a chaque caractère, d'en extraire la description (de la même façon que pour l'entraînement), et de faire une requête au modèle, pour connaître la classe la plus probable.

Le découpage des caractères est la phase la plus délicate. Pour l'effectuer, une solution consiste à seuiller l'image, pour avoir les lettres en blanc, et le fond en noir. Une fois cela effectué, on récupère le rectangle englobant chaque composante, et on le découpe.

Etape 2a : Binariser l'image
L'idéal est d'avoir une composante connexe (forme blanche contiguë) par caractère. Pour obtenir cela, on peut jouer sur :

- la technique de seuillage : simple ou adaptatif
- les paramètres de la méthode de seuillage
- des érosions et dilatations, pour enlever les artefacts, ou reconnecter les traits



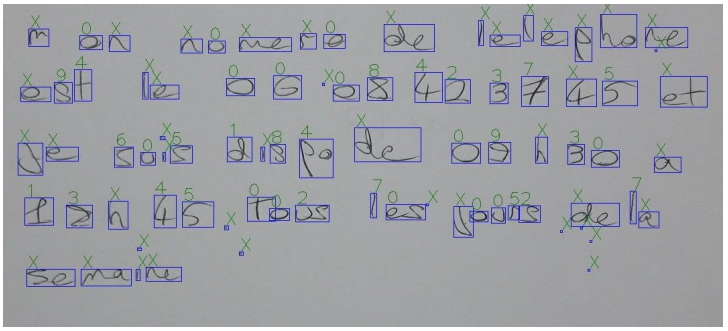


Etape 2b : Découper les zones d'intérêt
Une fois la binarisation effectuée, il faut extraire le caractère englobant de chaque caractère. Pour cela, vous pouvez utiliser la méthode floodFill qui permet de remplir une composante connexe, mais également de récupérer son rectangle englobant.
Une bonne technique pour cela peut consister a parcourir tous les pixels de l'image:

- si le pixel est blanc, remplir sa CC de noir, et ajouter son rectangle englobant a la liste

Pour chaque rectangle on peut ensuite facilement découper l'image correspondante en utilisant l'opérateur () de la classe Mat

Etape 2c : Prédire un résultat



Une fois chaque image de caractère extraite, on peut en extraire sa description, et faire une requête au modèle. Il ne reste plus qu'à afficher la classe récupérée à côté du rectangle correspondant.

AIDE : pour récupérer tous les fichiers d'un répertoire, sous linux, vous pouvez utiliser la fonction suivante :

```
#include <dirent.h> // for linux systems
#include <sys/stat.h> // for linux systems

/* Returns a list of files in a directory (except the ones that begin
with a dot) */
void readFilenames(std::vector<std::string> &filenames,
                   const std::string &directory)
{
    DIR *dir;
    class dirent *ent;
    class stat st;
    dir = opendir(directory.c_str());
    while ((ent = readdir(dir)) != NULL) {
        const std::string file_name = ent->d_name;
        const std::string full_file_name = directory + "/" +
file_name;
        std::cout<<full_file_name<<std::endl;
        if (file_name[0] == '.')
            continue;
        if (stat(full_file_name.c_str(), &st) == -1)
            continue;
        const bool is_directory = (st.st_mode & S_IFDIR) != 0;
        if (is_directory)
            continue;
        filenames.push_back(full_file_name); // returns just filename
    }
    closedir(dir);
}
```