

Reconnaissance des lettres de la langue américaine des signes

Le but de ce projet est d'utiliser un réseau de neurones (éventuellement à convolution) pour reconnaître les lettres de la langue américaine des signes à partir d'images.

1 Présentation du problème

La langue américaine des signes fait appel à 26 gestes de la main pour représenter chaque lettre (figure 1/bas). Les lettres J et Z nécessitent un mouvement et ne peuvent être reconnues uniquement à partir d'images. Un problème de discrimination des 24 positions de la main à partir d'images (figure 1/haut) a été proposé. Les données de ce challenge sont disponibles sur le site Kaggle¹. Elles correspondent à 27 455 images pour l'ensemble d'apprentissage et 7172 images pour l'ensemble de test. Chaque image a 28x28 pixels avec des valeurs en niveaux de gris entre 0-255 et est associée à une étiquette entre 0 et 25 correspondant à une lettre alphabétique A-Z (à l'exception de 9=J ou 25=Z).

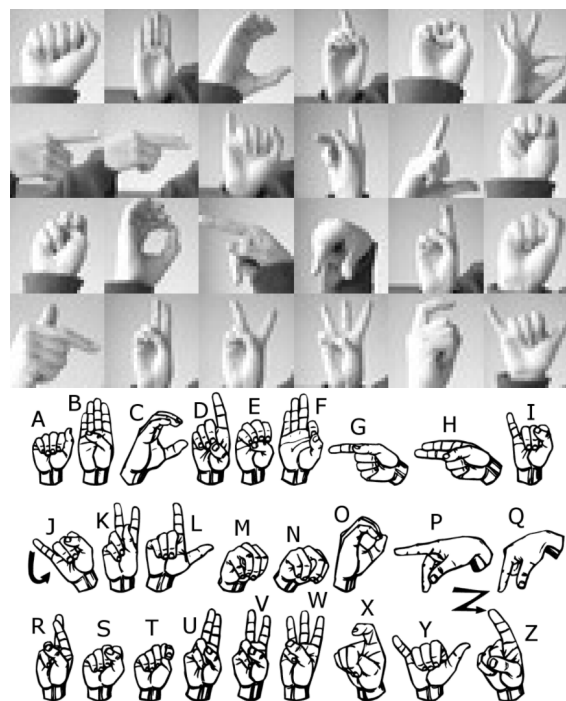


FIGURE 1 – Haut : exemples d'images de chaque lettre redimensionnées à 28x28 pixels en niveaux de gris avec des valeurs de pixel (0-255). Bas : Représentation dessinée des gestes de la main pour les lettres de la langue des signes américaine.

1. <https://www.kaggle.com/datamunge/sign-language-mnist>

2 Travail demandé

Le but du projet est de concevoir et d'entraîner un réseau de neurones (éventuellement à convolution) pour prédire la lettre signée en fonction d'une imagerie.

2.1 Réalisation

Les données et des détails sur le problème sont librement accessibles (<https://www.kaggle.com/datamunge/sign-language-mnist>).

Le choix du logiciel, de la bibliothèque ou du langage de programmation est libre. Cependant vous trouverez des informations pour vous guider sur l'utilisation de Keras et tensorflow en python à la fin du sujet.

Vous devrez présenter vos choix et les résultats de l'apprentissage automatique.

2.2 Présentation

Les soutenances auront lieu la semaine du 10 janvier 2022 en fonction de vos disponibilités aux uns et aux autres. Une présentation d'environ cinq minutes suivie d'une discussion d'une dizaine de minutes vous permettront d'une part d'exposer vos résultats en précisant leurs points forts, leurs points faibles, et les améliorations possibles et d'autre part d'effectuer une démonstration de l'implantation de cette stratégie. Le support de la présentation et les sources doivent être déposés sur le site arche dédié au module Intelligence Artificielle au plus tard le dimanche 9 janvier 2022. Merci de nommer votre archive avec le nom du projet et les vôtres (ex. LSA-Nom1-Nom2-Nom3.zip).

Aide à l'utilisation de réseaux profonds

2.3 Installation des bibliothèques avec PIP ou Anaconda

```
- numpy (pip install numpy ou / conda install numpy)
- sklearn (pip install -U scikit-learn ou conda install -c scikit-learn)
- scipy
- matplotlib
- opencv-python
- scikit-image
- tensorflow>=1.3.0
(pip install --upgrade tensorflow/conda install -c conda-forge tensorflow)
- keras>=2.0.8 (pip install keras/conda install -c conda-forge keras)
```

2.4 Vérification que les bibliothèques sont bien installées

```
import tensorflow as tf
from tensorflow.python import keras
print("Keras:", keras.__version__)
print("Tensorflow:", tf.__version__)
```

2.5 Liens utiles

Cliquez sur les liens.

- Anaconda :

<https://docs.anaconda.com/anaconda/install/index.html>

- keras :

<https://keras.io/>

- Le model Sequential (premier pas)

<https://keras.io/getting-started/sequential-model-guide/#getting-started-with-the-keras-sequential-model>

- Convolutional Neural Networks (CNNs) on Keras

<https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5>