Projet de fin d'étude

Analyse comparative des modèles de reconnaissance de posture

Introduction

Au regard de la littérature scientifique actuelle dans le domaine de la reconnaissance de posture, nous pouvons discerner deux catégories dominantes d'intelligence artificielle. L'une regroupe les modèles extrayant en amont le squelette, obtenu après un prétraitement de l'image, de la personne d'intérêt, l'autre englobe les méthodes de classification sans cette extraction ; dans les premières, nous retrouvons, en autres, les méthodes de *Support Vector Machine (*SVM) et les *K-Nearest Neighbors* (KNN) [1] [2], dans les secondes, sont compris les réseaux de neurones[4] [5] [6] et les Random Forest[3].
Notre choix s'est porté sur les réseaux de neurones pour différentes contraintes de temps, de ressources. Après quelques essais, nous avons décidé d'intégrer des modèles déjà entraînés sur la base de données ImageNet parmi nos couches pour augmenter la précision de détection.
Le présent document a pour but de récapituler les essais d'architecture les plus pertinents, dont voici la liste :

1) Un premier réseau de neurones relativement simples (quelques couches)
2) Un second intégrant un ResNet-50 [4]
3) Un troisième modèle couplant un VGG-16 [5] et quelques nouvelles couches
4) Un dernier utilisant un Inception-V3 [6] auquel nous ajoutons également quelques couches

Nous y décrirons les architectures, les résultats obtenus lors de la classification et justifierons notre choix final en nous basant sur les résultats comparatifs.

1) Un ResNet-50 [1] avec ajout d'une couche, repose sur une architecture plus complexe de CNN avec des skip connexion, ce réseau à montrer de bonne performance pour la classification d'image.
2) Un VGG-16, nous avons aussi implémenté ce modèle car dans la littérature scientifique ce modèle à été utilisé pour la reconnaissance de gesture[2].
3) Inception-V3[3], ce modèle est souvent utilisé pour l'analyse d'image et la détection d'objet est obtient de meilleures performances que les architectures VGGNet, ImageNet, GoogleNet.

Présentation de certains des modèles expérimentés

Avant d'entrer dans le détail des modèles, il est important de mentionner que chacune des architectures prend en entrée une image de taille 224*224*3. Tous les modèles ont été entraînés avec 40 epochs ou plus. La fonction d'activation en sortie est toujours une sigmoïde.

1) CNN

En premier lieu et avant d'entamer la construction de réseau de neurones plus complexes, nous avons envisagé sa simplicité. L'architecture consiste en 3 blocs séquentiels composés tous trois d'une couche de convolution 2D, suivie par une opération de normalisation par *batch*, puis une couche de *Max pooling*. Une couche de 5 neurones s'additionne en sortie pour produire la probabilité d'appartenir à chacune des classes.

Les couches de *max pooling* et de *normalisation par batch* sont utilisées pour éviter le sur-apprentissage et ainsi pour ne retenir que les caractéristiques les plus fortes de l'image. De plus, cela réduit le nombre de paramètres à optimiser et donc le temps d'apprentissage de nos modèles.

Durant la phase d'entraînement, nous avons obtenu de très bons résultats avec une *accuracy* de 100% sur la base d'apprentissage et de 96% sur la base de validation. Cependant, en pratique, la reconnaissance était mauvaise et ne donnait que 70,8% de bonnes détections pour la base de test, comme le montre les courbes.

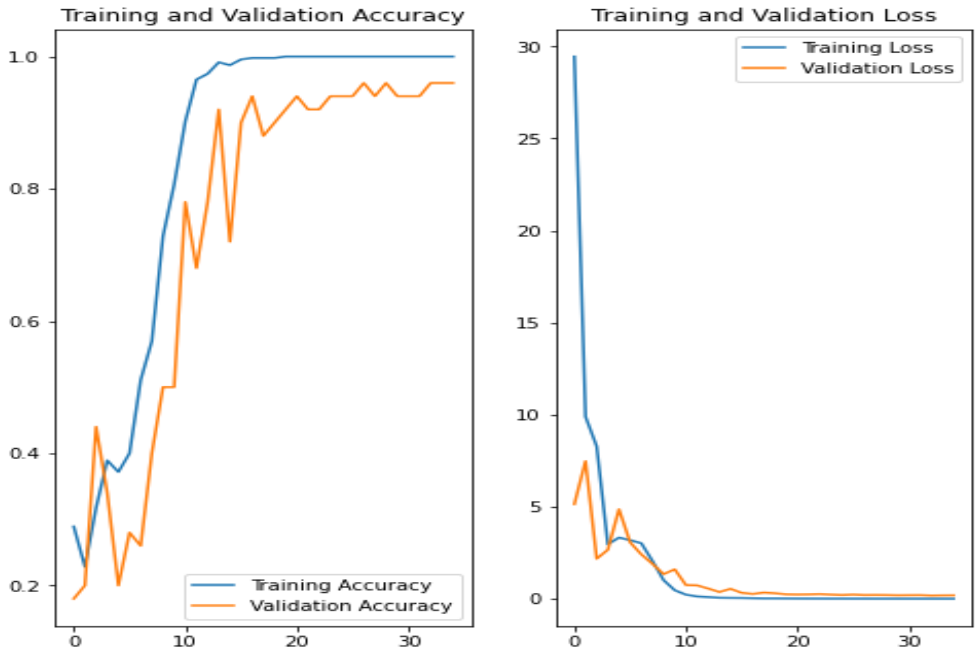

Figure 1 : Évolution de la précision du modèle CNN en fonction des epochs

2) ResNet-50

Après ces premiers résultats, nous avons décidé de détourner des modèles pré-entraînés auxquels nous avons ajouté des couches. Cette idée nous est venue des différents articles qui ont émergé lors des recherches de l'état de l'art. Après quelques tâtonnements, nous nous sommes tournés vers le ResNet-50 mis à disposition par Tensor Flow et pré-entraîné sur la base de données ImageNet. Nous avons élaboré l'architecture suivante :

- ResNet-50
- une opération de mise en forme de la sortie en vecteur avec du *dropout* (*flatten*)
- une couche de 1024 neurones avec un *dropout* de 0.5
- une couche de classification

Les couches de *flatten* sont utilisées pour obtenir des probabilités pour chacune des classes en sortie. De plus, après un premier essai sans, nous avons ajouté une couche intermédiaire de neurones *fully-connected* pour éviter une transition trop directe entre le masque de sortie du ResNet-50 et la classification en 5 poses. Cependant, en regard des résultats, il est possible que l'étape *flatten* et la couche de *fully-connected* soient totalement inadaptées à la résolution de notre problème : la précision sur la base d'apprentissage et de

la validation reste à 20% et n'augmente plus. Nous en concluons que la classification est totalement hasardeuse.

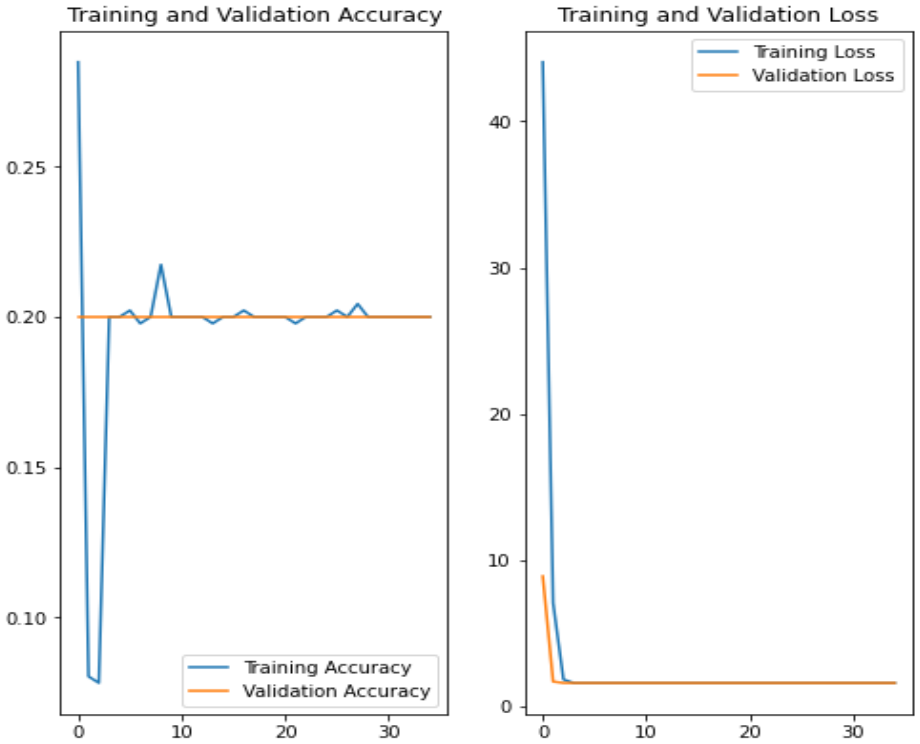

Figure 2 : Évolution de la précision du modèle 2 en fonction des epochs

3) VGG-16

Après les déboires de notre second modèle, nous avons opté pour l'intégration d'un VGG16 pré-entraîné sur la base de données ImageNet et provenant, à l'instar du ResNet-50, du *framework* Tensor Flow. À celui-ci, nous avons ajouté, de manière séquentielle, 3 blocs contenant une étape de normalisation par *batch* et une couche de neurones *fully connected*. En nous penchant sur les résultats du modèle sur la base de test, nous avons observé un sur-apprentissage du modèle à partir du 40ème épisode d'entraînement (*epoch*). Cependant, cette architecture, bien qu'elle offre des résultats de meilleure qualité pendant la phase de test (précision de 88%) est décevante lorsqu'il s'agit d'opérer la classification en temps-réel ; de nombreuses erreurs sont constatées pour la reconnaissance des poses 1 et 5, très souvent confondues.

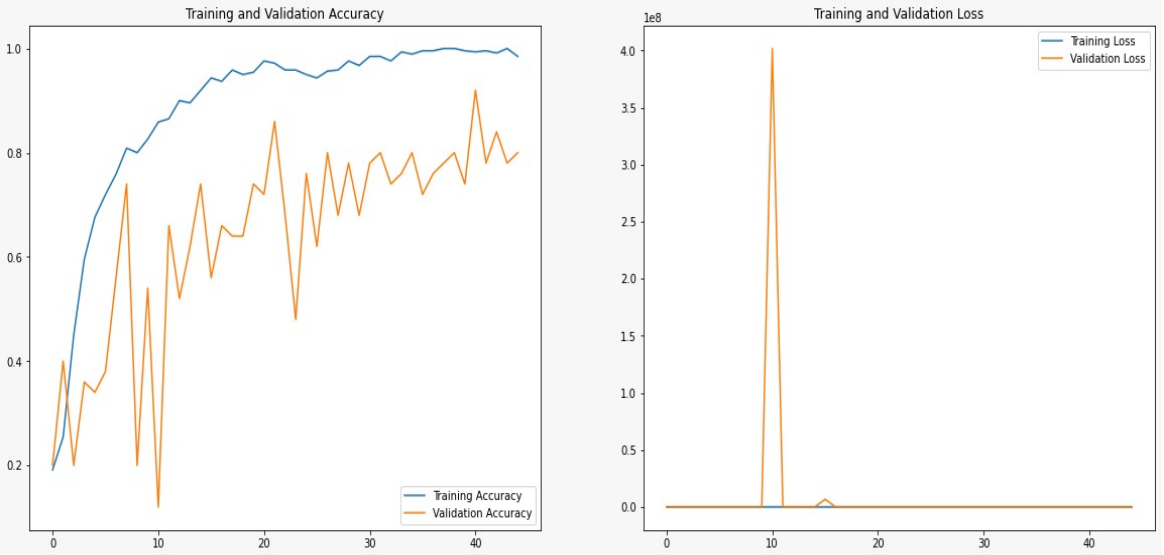

Figure 3 : Évolution de la précision du modèle 3 en fonction des epochs

4) Inception-V3

Après les échecs précédents, nous avons décidé d'approfondir la recherche dans l'état de l'art. Nous avons trouvé un article mentionnant le modèle d'*Inception V3*. Nous avons donc décidé, encore une fois, d'utiliser celui provenant de Tensor Flow pré-entraîné sur la base de données ImageNet auquel nous avons ajouté une première couches de neurones *fully connected*. Ensuite, afin d'améliorer les performances de classification, nous avons choisi d'ajouter :

- une couche de moyennage 2D (*global average pooling*), rendant le système plus robuste aux translations dans la sortie de l'*Inception V3*.
- une étape de *batch normalization* + une couche de *fully-connected*
- une étape de *batch normalization* + une couche de *fully-connected*

En réglant le *momentum* à 0.9 et le *learning-rate* à 0.001, nous avons atteint une précision satisfaisante de 88%, de 86% pour l'ensemble de test et qui s'est avérée stable lors du passage à la détection réelle. Toutefois, ponctuellement, les poses 1 et 5 sont confondues mais ce problème est mineur et est imperceptible lors du jeu.

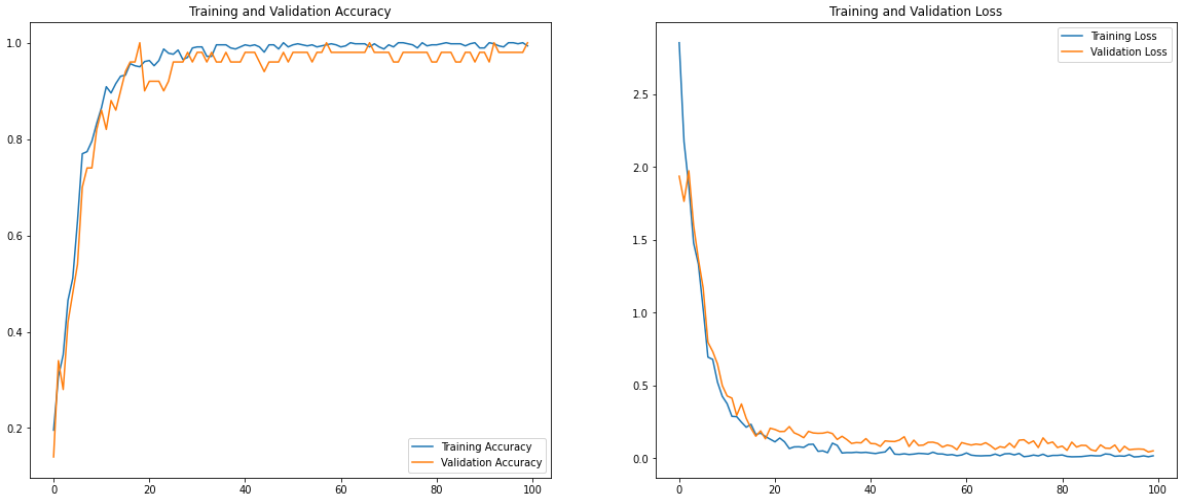

Figure 4 : Évolution de la précision du modèle 4 en fonction des epochs

Conclusion

Après avoir analysé les performances de chacun des modèles, nous avons retenu le dernier. Non seulement, la précision est la plus haute lors des essais de détection en temps-réel mais, en plus, ce modèle est le plus robuste aux changements de luminosité et à l'imprécision de la position d'une personne. Notre modèle est prometteur et est celui qui, de tous ceux essayés, apparaît comme le plus adapté à la reconnaissance de poses dans le contexte d'un jeu avec des enfants.

Bibliographie et état de l'art

[1] Classification of yoga pose using machine learning techniques, J.Palanimeera et K.Ponmozhi, https://doi.org/10.1016/j.matpr.2020.08.700
[2] Skeleton-based Dynamic hand gesture recognition Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre
[3] A Random Forest Approach to Segmenting and Classifying Gestures Ajjen Joshi , Camille Monnier , Margrit Betke and Stan Sclaroff
[4] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016
[5] Gesture Recognition using CNN and RNN Rajalakshmi J, Kumar P, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-9 Issue-2, July 2020
[6] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015