- Etat de l'art

- les articles

Avant de mettre en œuvre un plan d’action nous avons étudié quatre articles pour avoir une meilleure idée sur les solutions existantes et les résultats obtenus.

- choix du modèle

Nous avons choisi de travailler avec le modèle siamois pour sa compatibilité avec notre problème. En un réseau de neurones siamois (parfois appelé réseau neuronal jumeau) est un réseau de neurones artificiel qui utilise les mêmes poids tout en travaillant en tandem sur deux vecteurs d'entrée différents pour calculer des vecteurs de sortie comparables. Souvent, l'un des vecteurs de sortie est précalculé, formant ainsi une ligne de base par rapport à laquelle l'autre vecteur de sortie est comparé. Dans notre cas, une paire de signatures est donnée en entrée avec la première signature qui est toujours authentique est la deuxième qui est soit authentique soit falsifiée. Pour conclure sur la nature de la deuxième signature, un score de différence est calculé entre les deux vecteurs de sortie fournis par le modèle selon la formule de distance euclidienne et un seuil est déterminé par le modèle. Dans le cas où le score dépasse le seuil on décide que la signature est falsifiée et dans le cas contraire la signature est libellée comme authentique.

- Les bases de données

Nous avons découvert plusieurs bases de données contenant des signatures authentiques et falsifiées d’un nombre variable d’auteurs. Les bases de données sont assez variées quant à l’origine des signatures (Inde, Bengladesh, Europe) et donc leur style ainsi que dans la taille des images fournies. Les bases de données traitées dans ce projet sont :

♦ BHSig260 : Cette base de données contient des signatures authentiques et imitées pour 260 auteurs au total divisés entre des signatures Hindi (de l’Inde) et des signature Bengali (du Bengladesh).

♣ Hindi : Contient les signatures de 160 auteurs. Pour chaque auteur, nous avons 24 signatures authentiques et 30 signatures imitées.

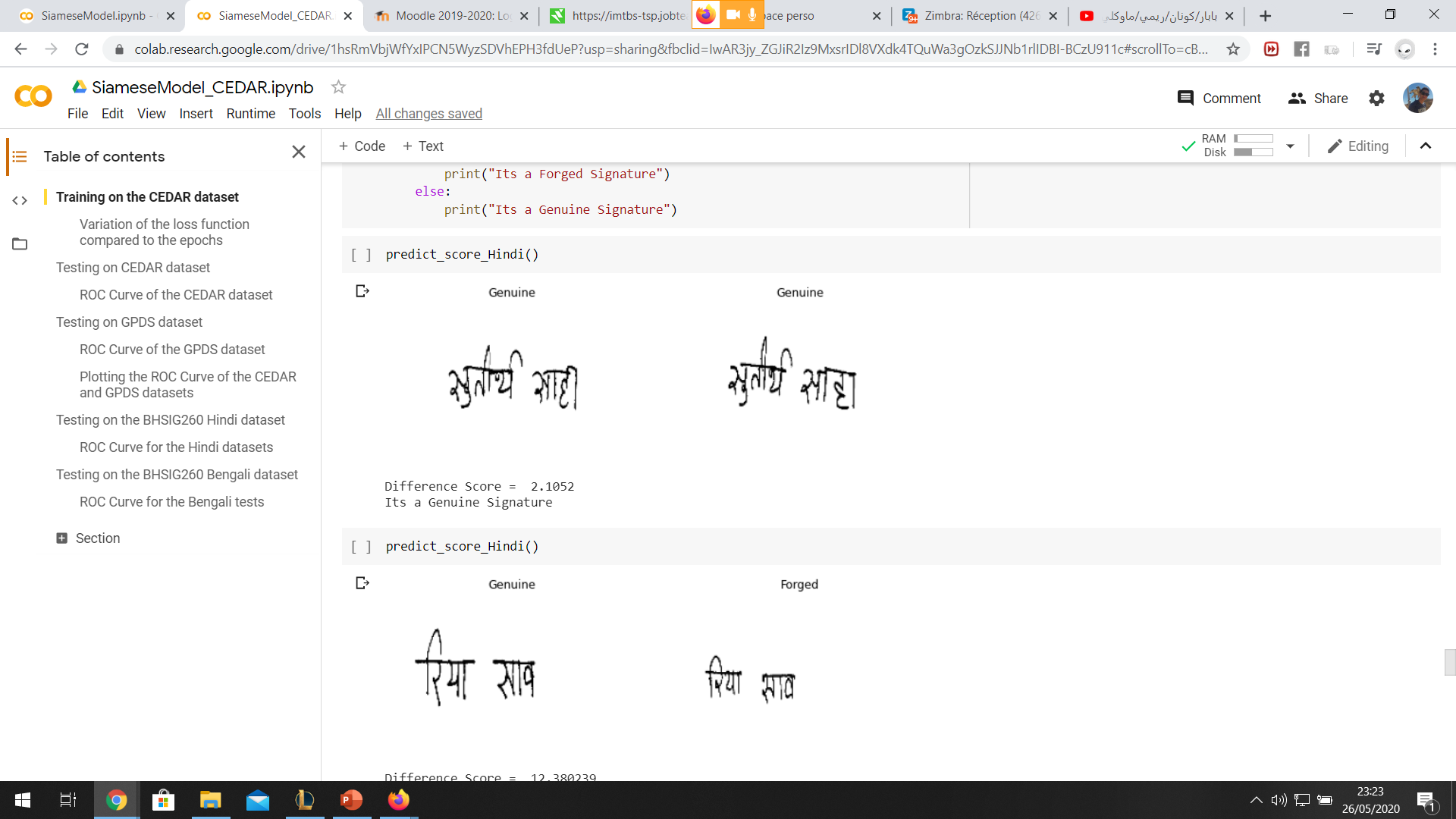


Figure 1 : Echantillon de paire de signatures (Genuine, Genuine) de la base BHSig260 : Hindi

♣ Bengali : Contient les signatures de 100 auteurs. Pour chaque auteur, nous avons 24 signatures authentiques et 30 signatures imitées.

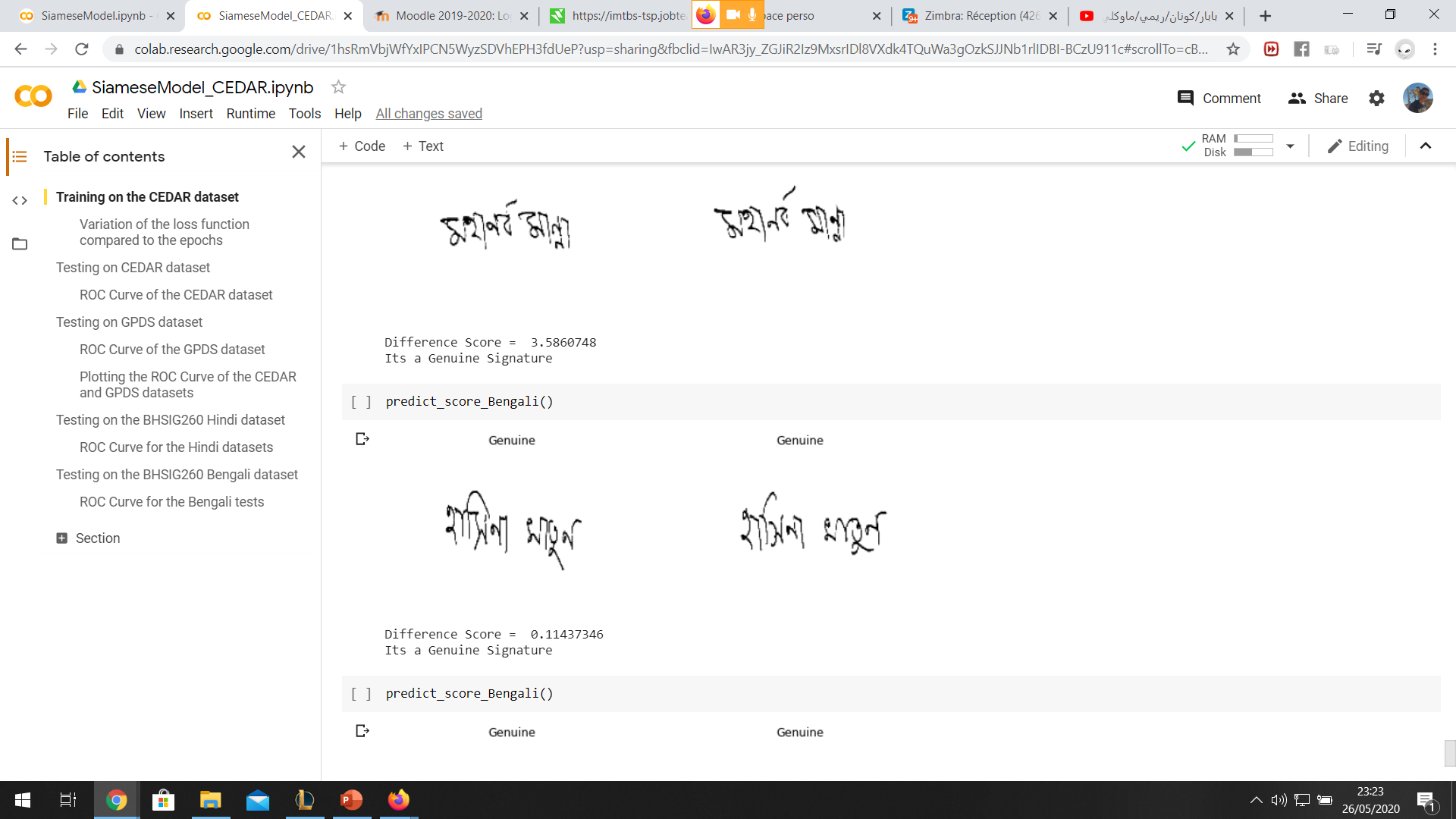


Figure 2 : Echantillon de paire de signatures (Genuine, Genuine) de la base BHSig260 : Bengali

♦ GPDS300 : Cette base de données contient des signatures européennes authentiques et imitées pour 300 auteurs. Pour chaque auteur nous avons 24 signatures authentiques et 30 signatures imitées.

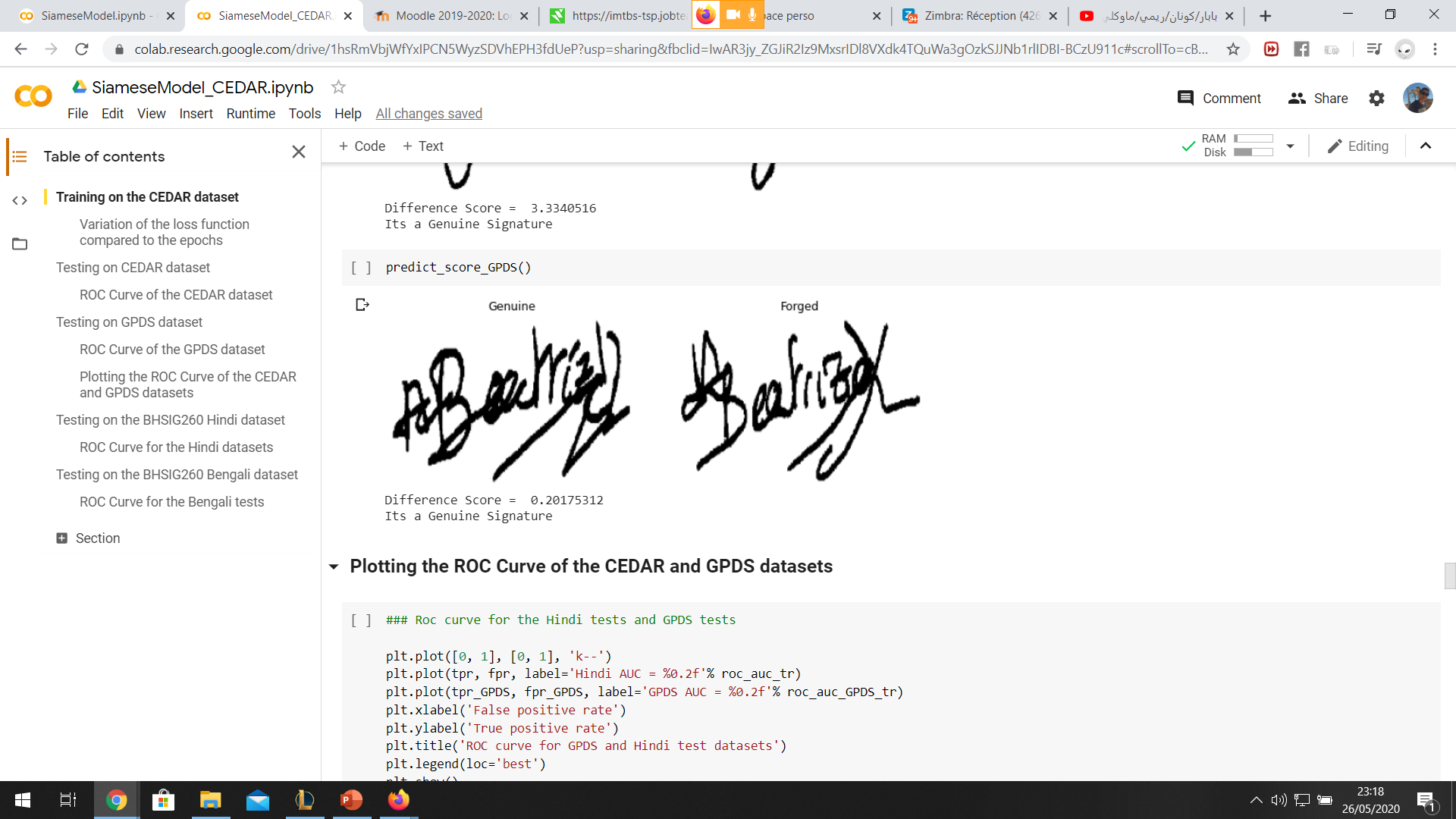


Figure 3 : Echantillon de paire de signatures (Genuine, Forged) de la base GPDS300

♦ CEDAR : Cette base de données contient des signatures européennes authentiques et imitées pour 55 auteurs. Pour chaque auteur nous avons 24 signatures authentiques et 24 signatures imitées.

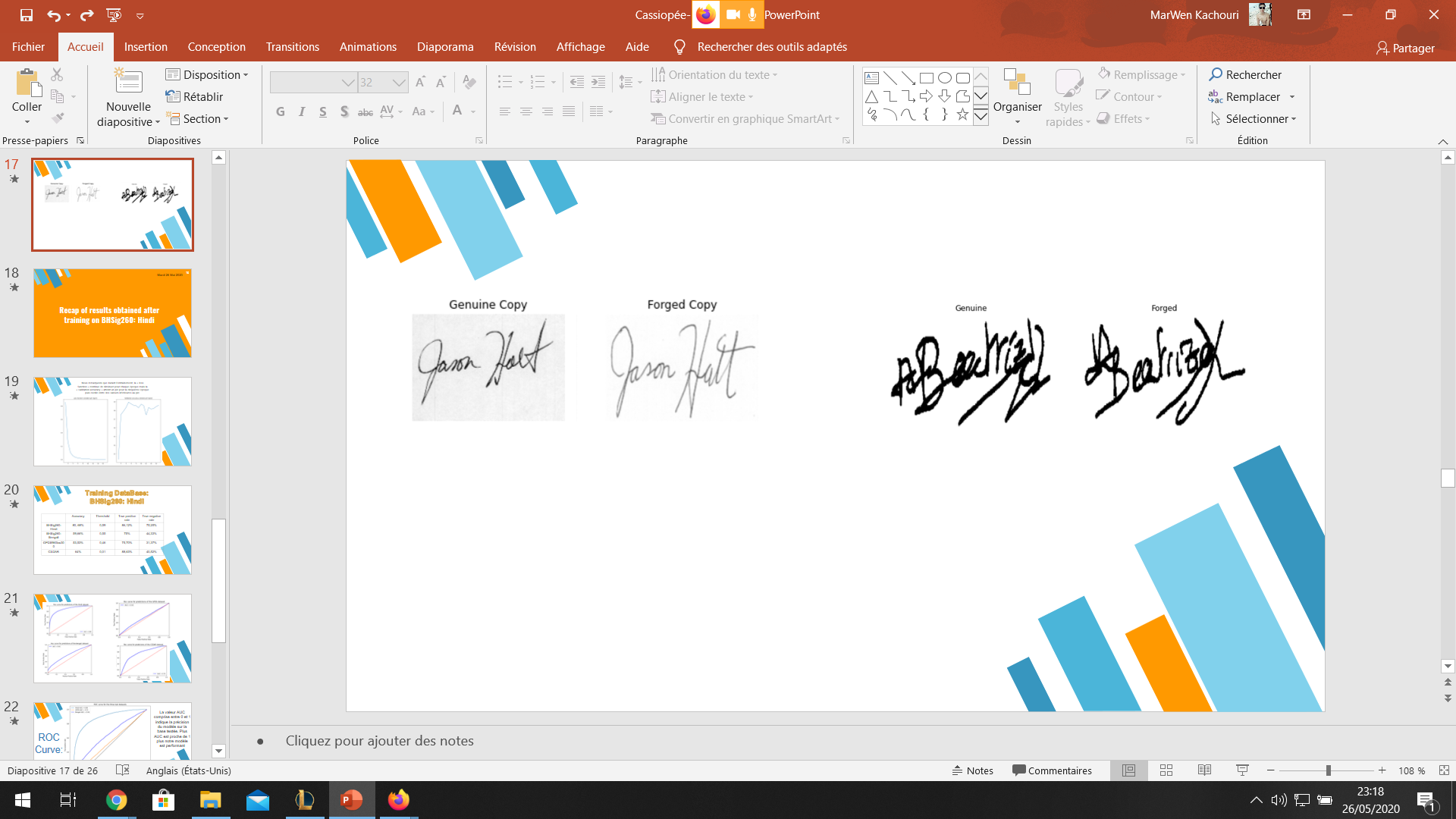
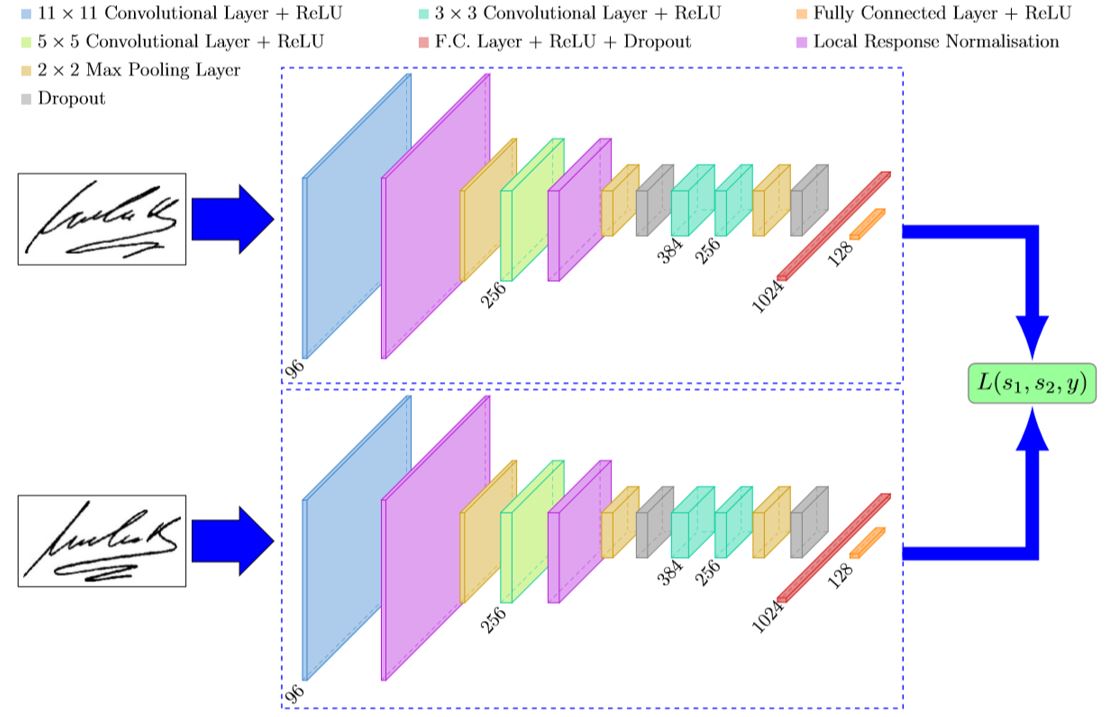


Figure 4 : Echantillon de paire de signatures (Genuine, Genuine) de la base CEDAR

- D’autres bases de données ont été abordées à l’étude d’article mais non traitées telles que DS-GET, GPDS Synthetic, SigComp etc.

- Implémentation de la solution

 - Architecture du modèle :

- Comme représenté dans la figure ci-dessus, nous avons deux réseaux identiques qui prennent en entrée une signature afin de calculer la différence de score entre les deux vecteurs de sorties.

- Convolutional Layer : représente une couche de convolution dans le réseau. On réalise en succession l’opération de convolution de l’image en entrée avec un filtre de taille précisée pour chaque couche (11x11, 3x3, 5x5).

- ReLU : Ou Rectified Linear Unit est une fonction d’activation ayant pour expression :

f(x) = max(0,x)

- 2x2 Max pooling : Pour chaque sous matrice de taille 2x2 le maximum des 4 pixels est choisi comme valeur de sortie. L’objectif de cette couche est de sous-échantillonner l’image en réduisant sa dimension.

- Fully Connected Layer : c’est une couche où toutes les entrées d'une couche sont connectées à chaque unité d'activation de la couche suivante.

- Dropout : On ignore des connections avec une probabilité p. C’est une technique utilisée pour résister au phénomène de sur-apprentissage. Dans notre cas la probabilité de Dropout p= 0.3 lorsque la couche de Dropout suit celle de Max pooling et p=0.5 lorsqu’on suit une Fullt Connected Layer.

- Local Response Normalisation : C’est une opération de normalisation sur les valeurs de pixels de l’image en entrée.

- pré-traitement

- Dans le but de maximiser les performances de notre modèle un pré-traitement sur les images d’entrées s’impose. Dans notre cas notre pré-traitement se présente sous la forme de rogner les images selon une taille fixe étudiée : 155x220. Ce pré-traitement nous permet d’introduire moins de facteurs de bruit sur le modèle afin qu’il apprenne les vraies caractéristiques d’une signature imitée.

- Entrainement

Principe :

- Nous entrainons le modèle sur des paires de signatures « Genuine,Genuine » et « Genuine,Forged ».

- Le modèle est compilé avec la fonction de perte « Contrastive Loss ».

- À l'aide des « Keras Callbacks », nous enregistrons les poids obtenus par le modèle après chaque époque.

- Nous Réduisons le taux d'apprentissage d'un facteur 0,1 si la perte de validation ne s'améliore pas pendant 5 époques.

- Nous arrêtons l’entrainement en utilisant un arrêt précoce si la perte de validation ne s'améliore pas pendant 12 époques.

Spécificités pour les bases de données d’entrainement :

- Nous avons fait le choix d’entrainer le modèle sur les bases de données BHSig260 : Hindi et CEDAR.

♦BHSig260 : Hindi : Nous entrainons le modèle sur les signatures de 120 auteurs aléatoires parmi les 160. Pour chaque auteur, nous prenons la totalité des signatures authentiques, soit 24 signatures, et 12 signatures imitées aléatoirement tirées des 30 existantes. Donc, pour chaque auteur nous avons paires « Genuine, Genuine » possibles et **12\*24 = 288 paires** « Genuine, Forged » possibles. Au total nous avons un nombre d’échantillons égal à **120\*(276 + 288) = 67 680.**

♦CEDAR : Nous entrainons le modèle sur les signatures de 35 auteurs aléatoires parmi les 55. Pour chaque auteur, nous prenons la totalité des signatures authentiques, soit 24 signatures, et 12 signatures imitées aléatoirement tirées des 24 existantes. Donc, pour chaque auteur nous avons paires « Genuine, Genuine » possibles et **12\*24 = 288 paires** « Genuine, Forged » possibles. Au total nous avons un nombre d’échantillons égal à **35\*(276 + 288) = 19 740.**

- Les résultats de l’entrainement sont présentés dans la section Résultats.

- bout de code fih architecture w training

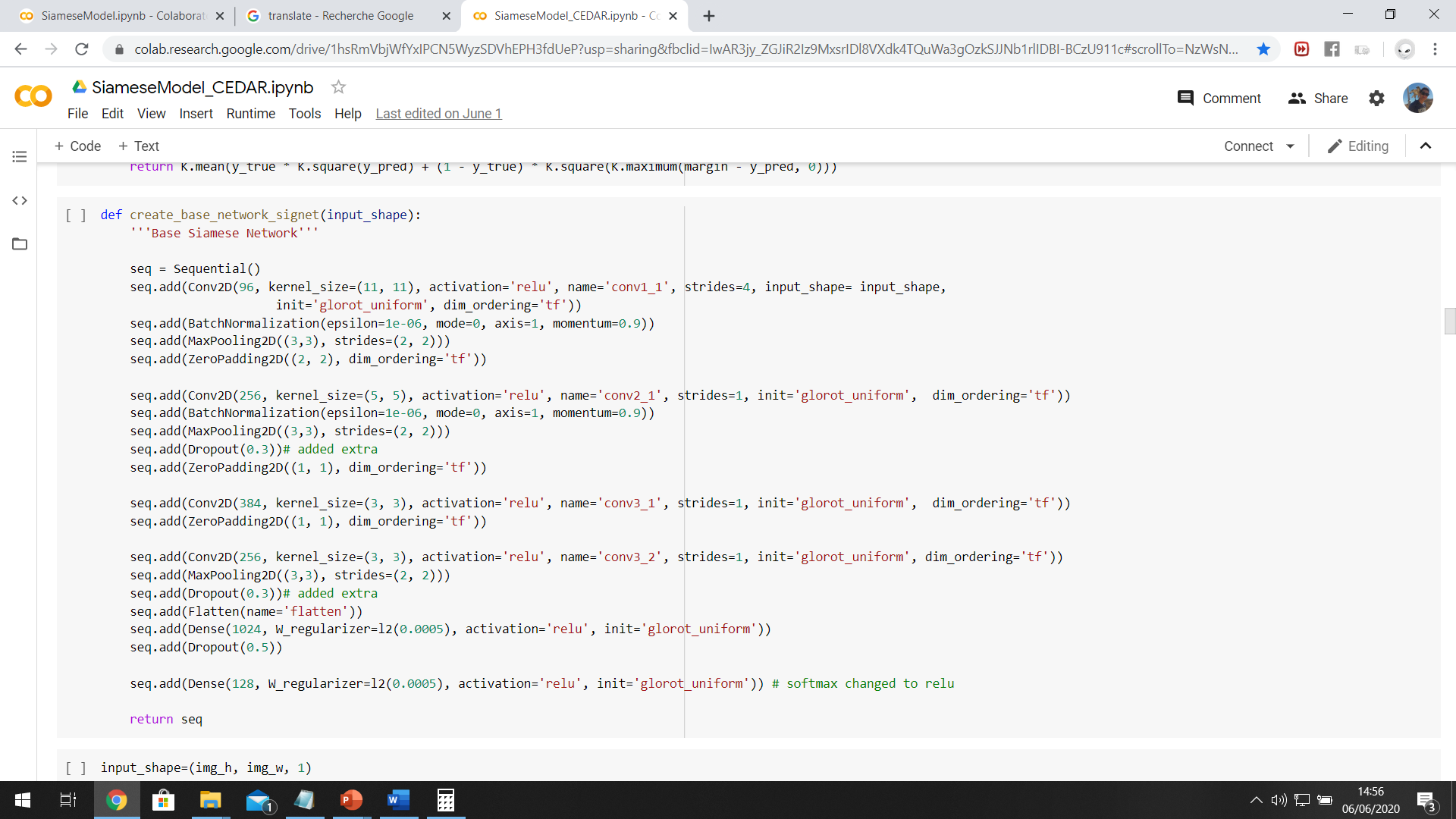
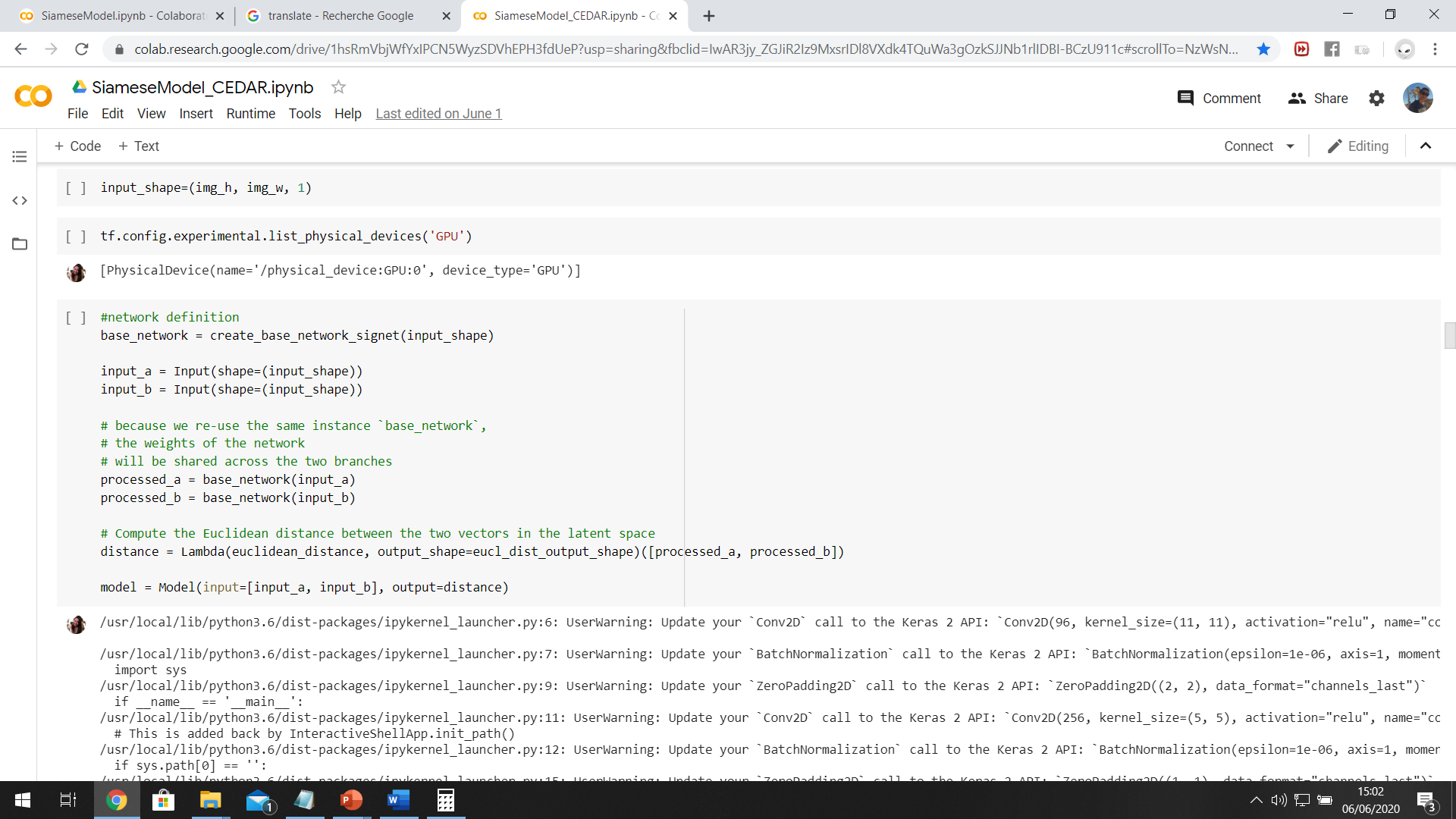


Figure : Lignes de code pour la définition de l’architecture du réseau



Deux sous-réseaux identiques conformément à la définition d’un réseau siamois

Les deux vecteurs de sortie

Calcul de la différence de score (distance euclidienne)

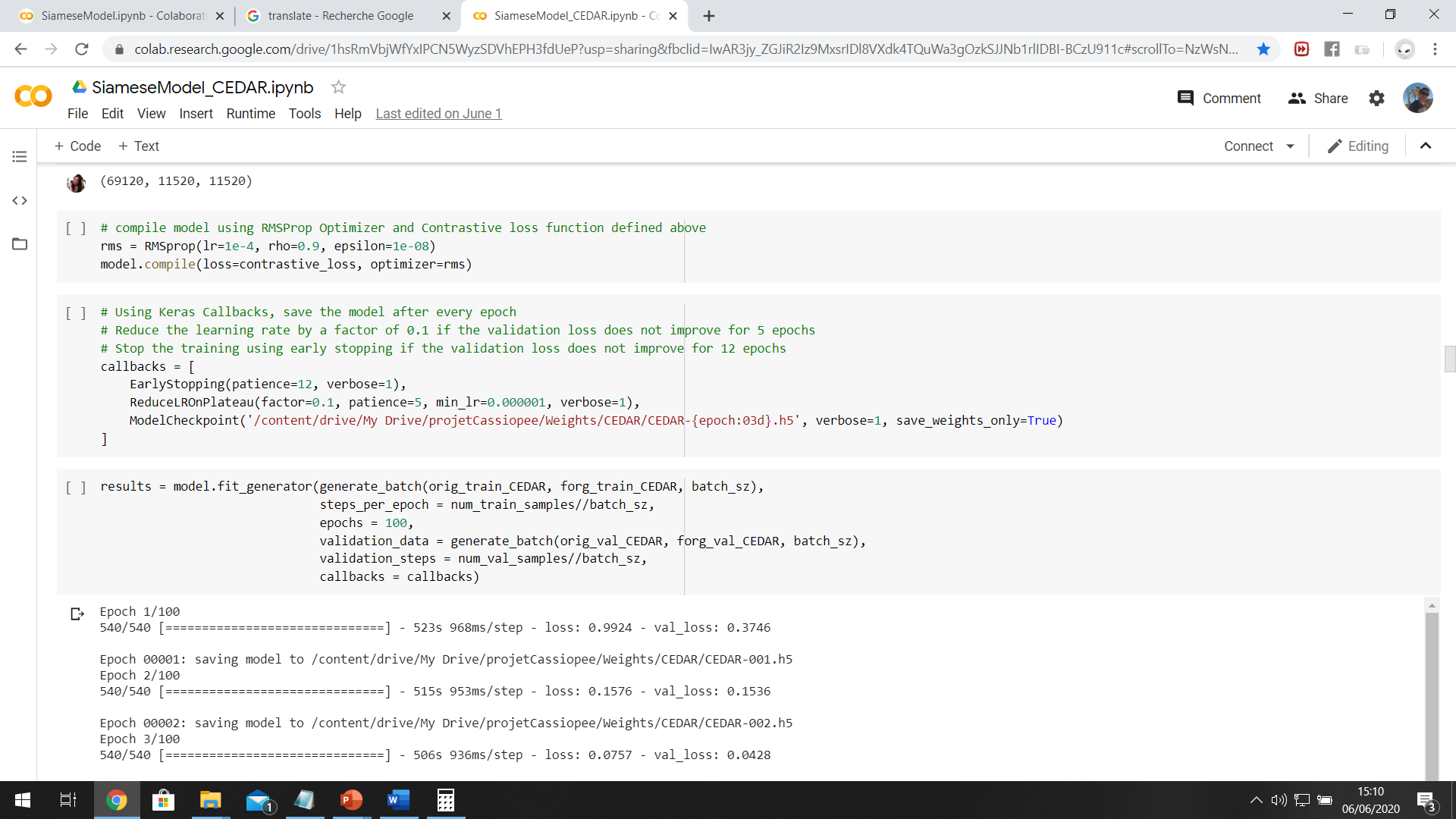
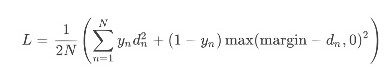


Figure : Entrainement du modèle sur la base de données CEDAR

- Loss function

- Conformément aux choix pris dans l’article étudié concernant le réseau de neurones Siamois, nous avons choix la fonction de perte « Contrastive Loss » ayant pour formule :

Avec yn est la sortie réelle, dn est la sortie prédite par le modèle, margin est la marge et N le nombre d’échantillons.