

# Reconnaissance des lettres de la langue américaine des signes

**Ulysse MARCANELLA**  
**Lucas RIOUX**

# SOMMAIRE

01

## CONTEXTE

Objectifs du projet, jeu de données

02

## PERCEPTRON MULTICOUCHE

Première approche du problème

03

## RESEAU CONVOLUTIF

Amélioration des résultats

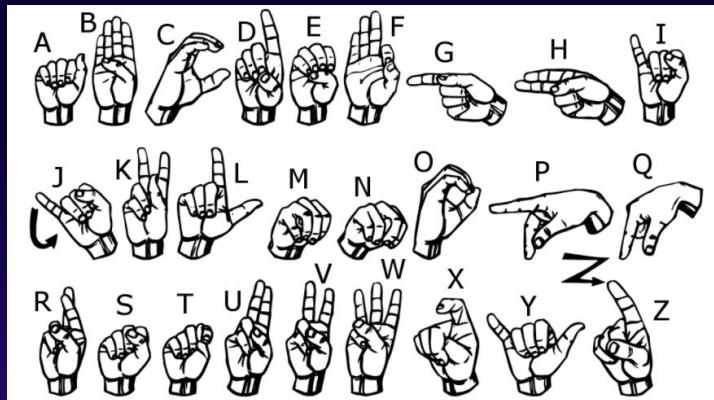
04

## INTERPRETATION

Analyse des résultats et voies d'amélioration

# CONTEXTE

Objectif : reconnaître les lettres de la langue des signes américaine



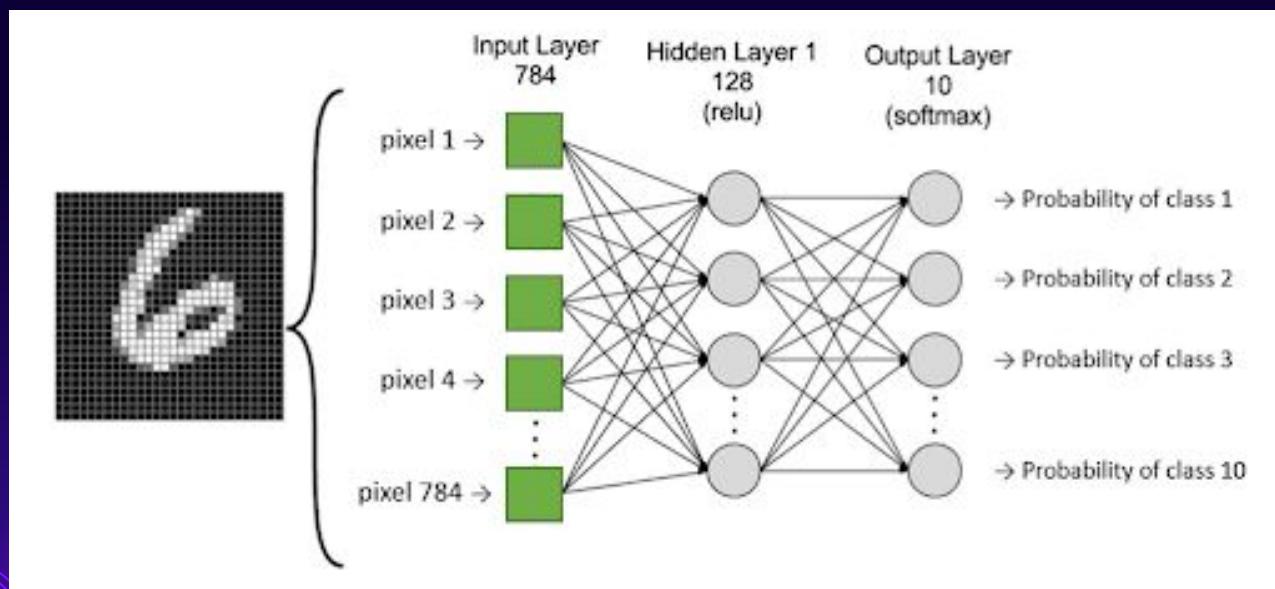
Lettres de la langue des signes américaine



Jeu de données utilisé (28 x 28)  
- 27 455 images d'entraînement  
- 7 172 images d'évaluation

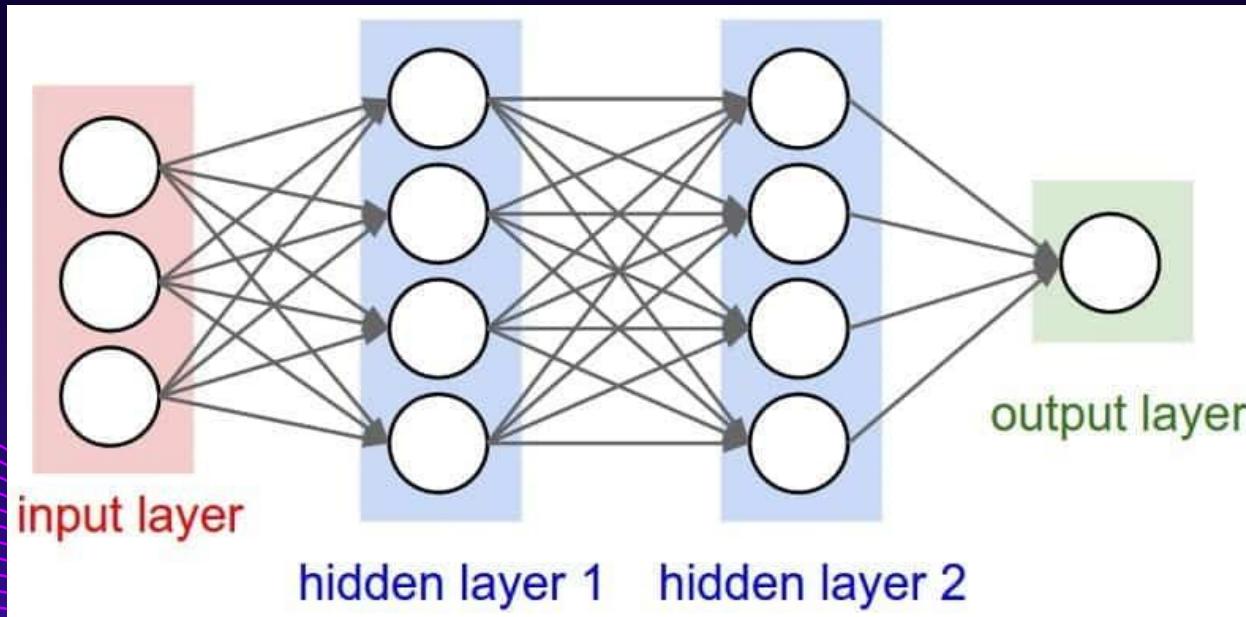
# RÉSEAU DE NEURONES

Modèle mathématique permettant de **classifier** des données



# PERCEPTRON MULTICOUCHES

Plusieurs couches de neurones **superposées entièrement connectées**



# MODÈLE

```
model = Sequential(name="perceptron")  
  
model.add(Flatten(input_shape=(28, 28)))  
model.add(Dense(512, activation="relu"))  
model.add(Dense(256, activation="relu"))  
model.add(Dense(128, activation="relu"))  
model.add(Dense(64, activation="relu"))  
model.add(Dense(26, activation="softmax"))
```

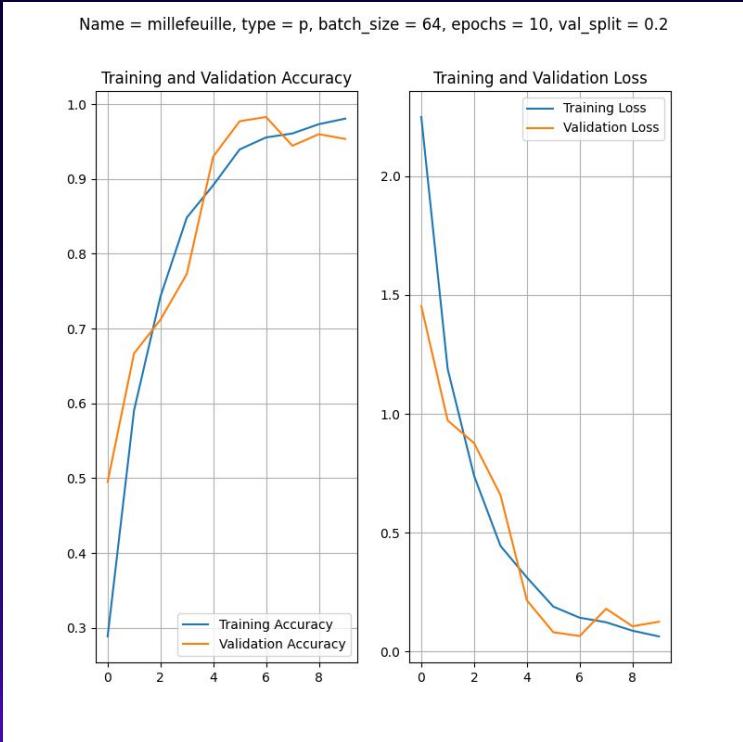
Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 512)	401920
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 26)	1690
<hr/>		
Total params: 576,090		
Trainable params: 576,090		
Non-trainable params: 0		

# RÉSULTATS

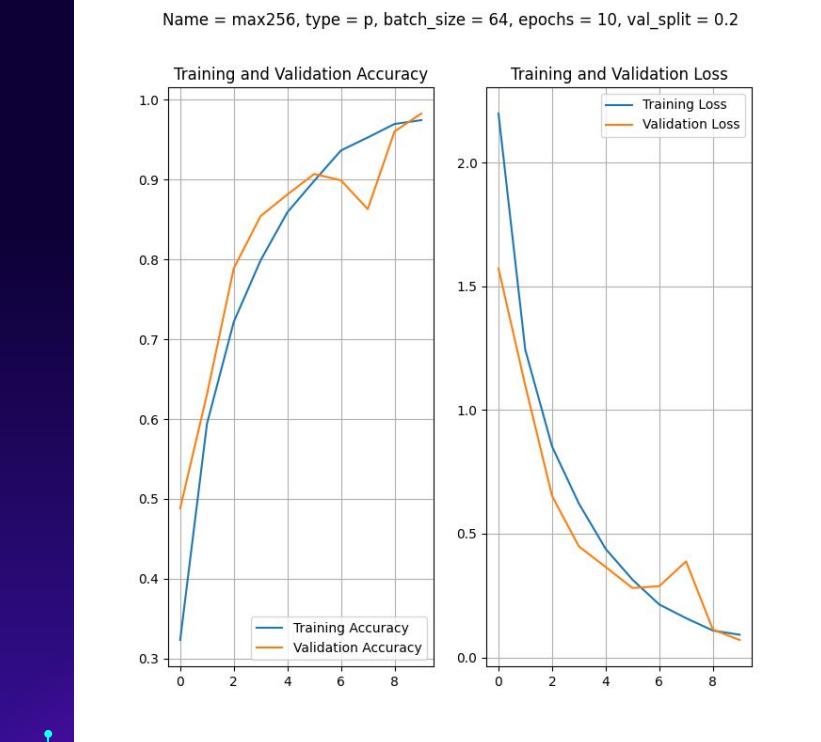
Modèle	512 - 256 - 128 - 64 - <b>26</b>	256 - 128 - 64 - <b>26</b>
Précision maximum	<b>81,62 %</b>	<b>76,60 %</b>
Précision moyenne	73,56 %	73,60 %
Écart-type de la précision	4,57 %	2,37 %
Perte moyenne	1,28	1,14
Écart-type de la perte	0,24	0,15
Temps	5:27 (32,7 s / ité.)	6:53 (41,3 s / ité.)

# GRAPHIQUES

512 - 256 - 128 - 64 - **26**

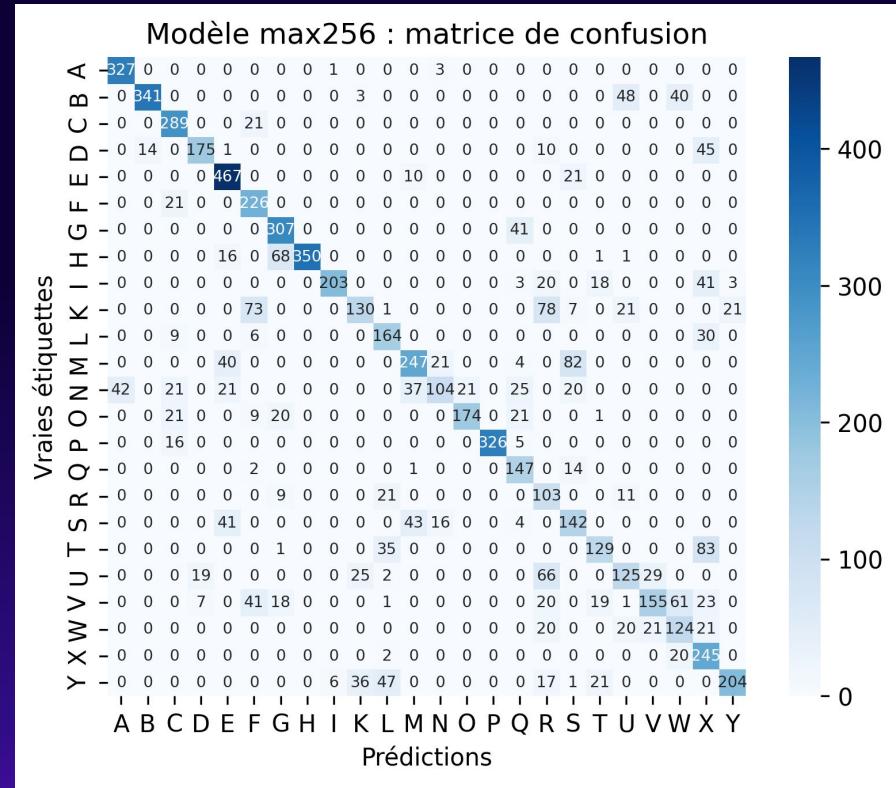
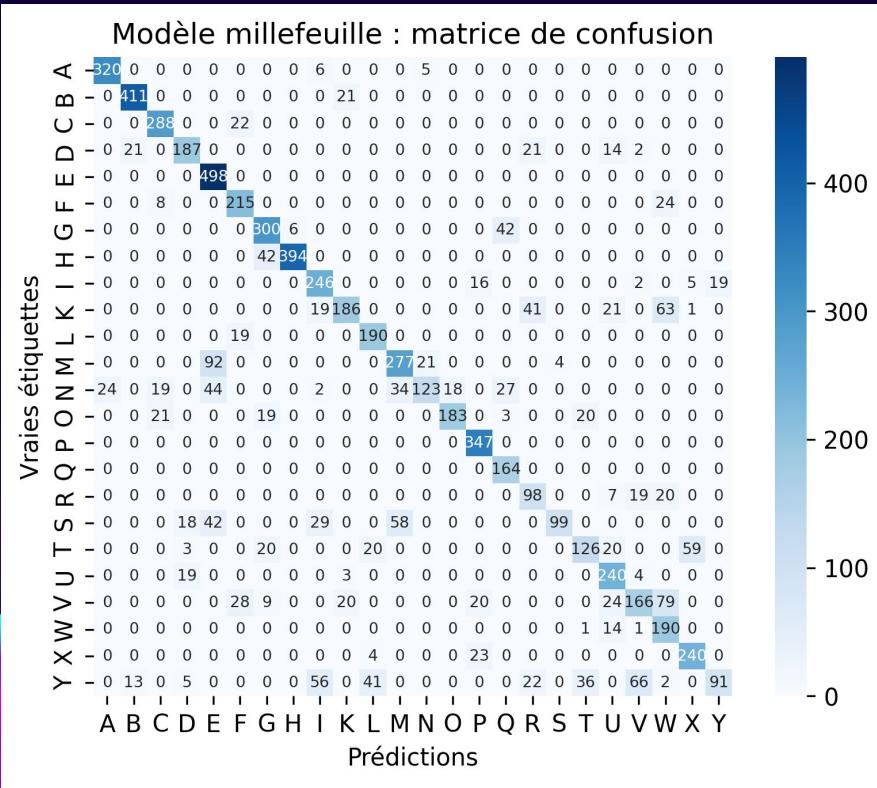


256 - 128 - 64 - **26**



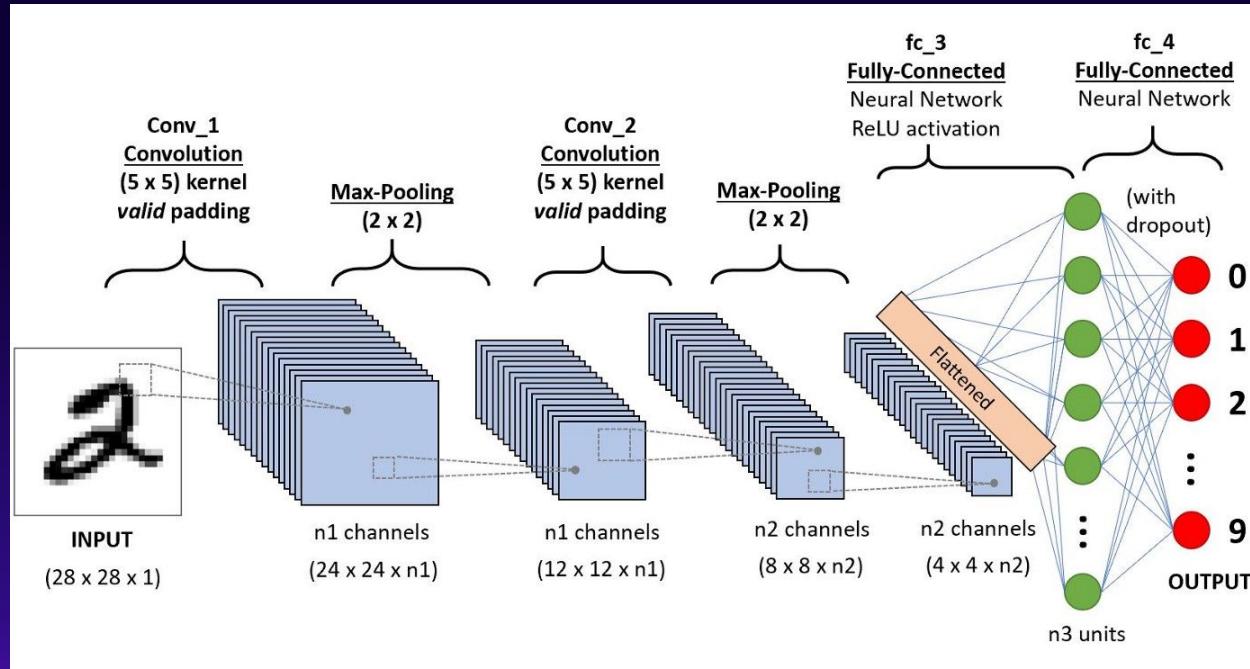
# MATRICES DE CONFUSION

512 - 256 - 128 - 64 - **26**



# RÉSEAU CONVOLUTIONNEL (CNN)

Architecture spécialement conçue pour la **classification d'images**



# MODÈLE

```
model = Sequential(name="convolutional")

model.add(Conv2D(48, (5, 5), activation="relu", input_shape=(28, 28, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))

model.add(Conv2D(72, (3, 3), activation="relu"))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.55))

model.add(Flatten())
model.add(Dense(512, activation="relu"))
model.add(Dense(512, activation="relu"))
model.add(Dense(256, activation="relu"))
model.add(Dense(256, activation="relu"))
model.add(Dense(128, activation="relu"))
model.add(Dropout(0.7))

model.add(Dense(26, activation="softmax"))
```

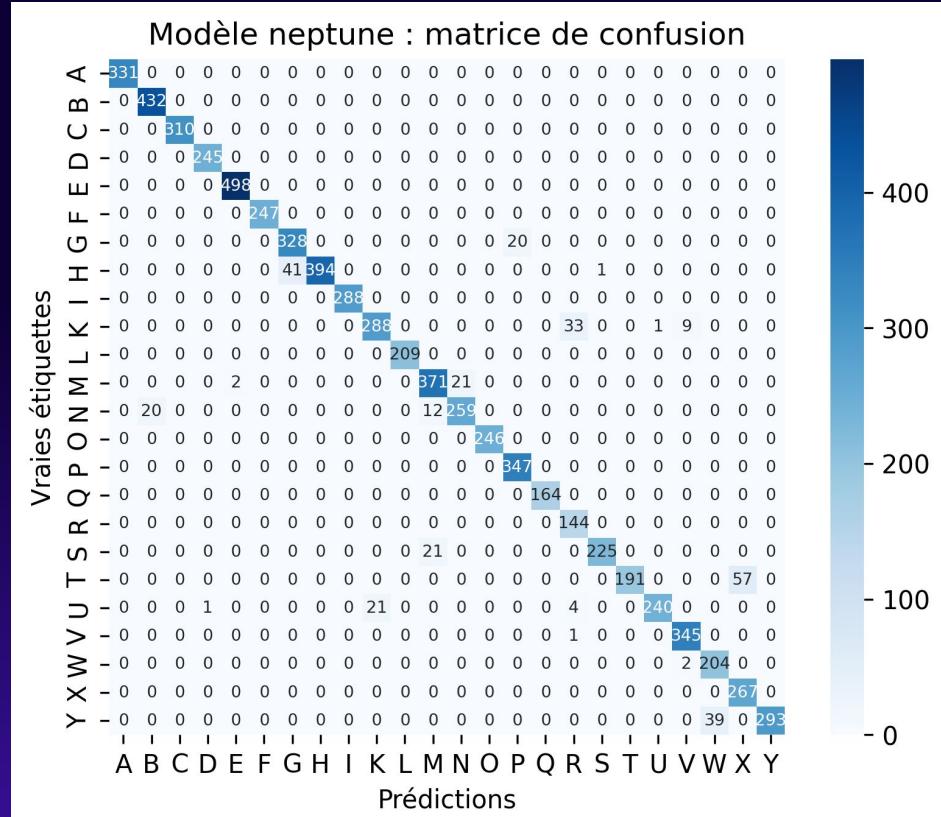
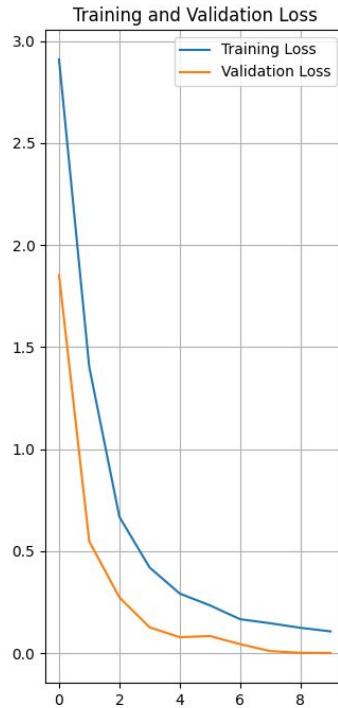
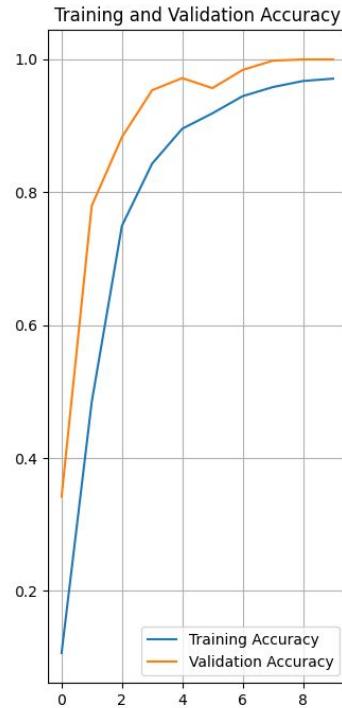
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 24, 24, 48)	1248
max_pooling2d (MaxPooling2D)	(None, 12, 12, 48)	0
dropout (Dropout)	(None, 12, 12, 48)	0
conv2d_1 (Conv2D)	(None, 10, 10, 72)	31176
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 72)	0
dropout_1 (Dropout)	(None, 5, 5, 72)	0
flatten (Flatten)	(None, 1800)	0
dense (Dense)	(None, 512)	922112
dense_1 (Dense)	(None, 512)	262656
dense_2 (Dense)	(None, 256)	131328
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 26)	3354
Total params: 1,450,562		
Trainable params: 1,450,562		
Non-trainable params: 0		

# RÉSULTATS

<b>Modèle</b>	512 - 256 - 128 - 64 - <b>26</b>	Réseau convolutionnel
<b>Précision maximum</b>	<b>81,62 %</b>	<b>96,60 %</b>
<b>Précision moyenne</b>	73,56 %	95,28 %
<b>Écart-type de la précision</b>	4,57 %	1,17 %
<b>Perte moyenne</b>	1,28	0,26
<b>Écart-type de la perte</b>	0,24	0,09
<b>Temps</b>	5:27 (32,7 s / ité.)	35:33 (213,3 s = 3:33 / ité.)

# GRAPHIQUE ET MATRICE DE CONFUSION

Name = neptune, type = c, batch\_size = 64, epochs = 10, val\_split = 0.2





MERCI DE VOTRE  
ATTENTION !

# SOURCES

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

Thème du diaporama :

<https://slidesgo.com/theme/dynamic-data-architecture-plan#search-data&position-32&results-65&rs=search> [Crédits diapositive suivante]

Images :

- <https://datascientest.com/fonctionnement-des-reseaux-neurones>
- <https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition>
- <https://paperswithcode.com/methods/category/convolutional-neural-networks>

Développement :

- <https://keras.io/getting-started/sequential-model-guide/#getting-started-with-the-keras-sequential-model>
- <https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbad5f5>
- <https://www.tensorflow.org/tutorials/quickstart/beginner?hl=fr>
- <https://www.tensorflow.org/tutorials/images/classification?hl=fr>

# THANKS!

Do you have any questions?

[youremail@freepik.com](mailto:youremail@freepik.com)

+91 620 421 838

[yourcompany.com](http://yourcompany.com)



CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#) and infographics & images by [Freepik](#)

Please keep this slide for attribution