

# Projet Deep Learning



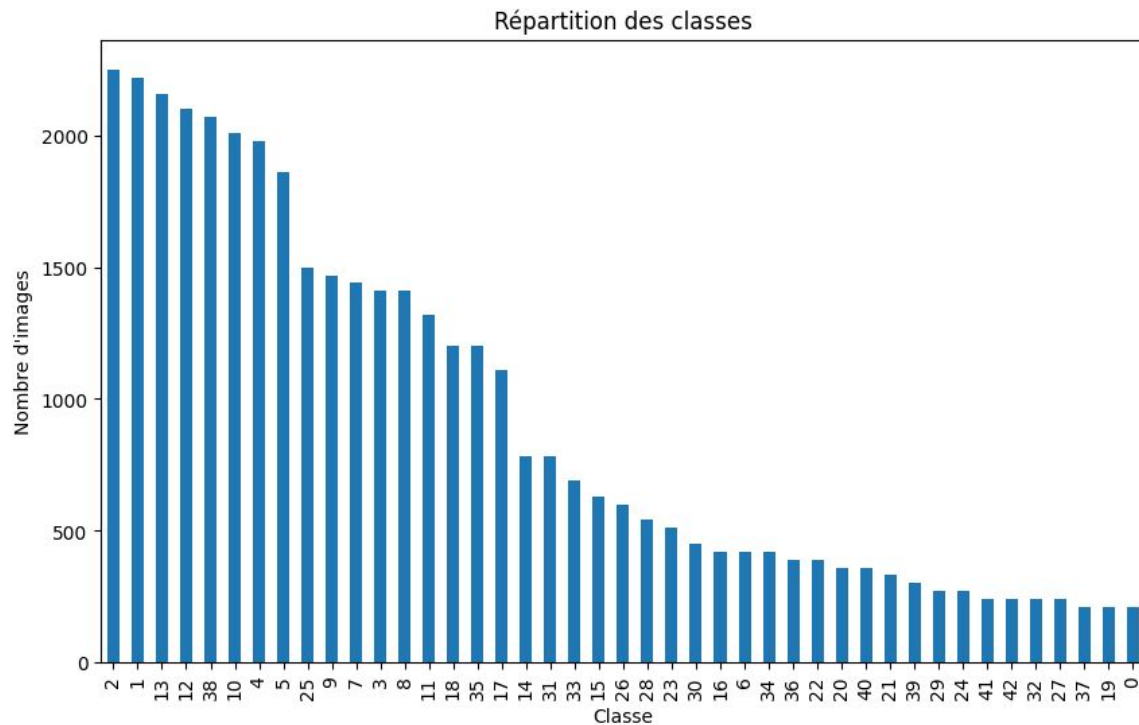
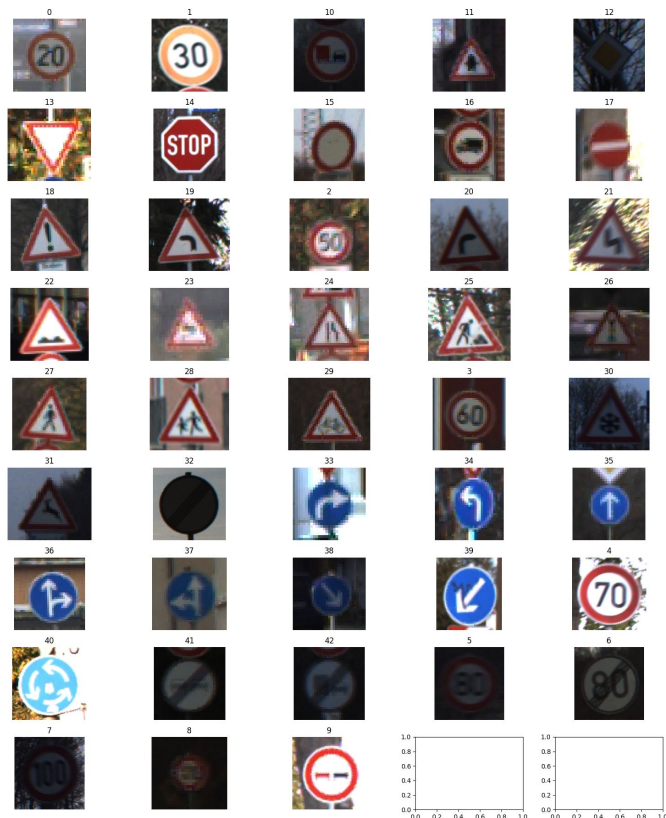
## Membre du groupe :

1. Nadia Medjdoub
2. Karim Ameer

# Introduction

- La reconnaissance des panneaux de signalisation est cruciale pour la perception et la prise de décision des véhicules autonomes.
- Deux aspects clés : perception (identification des objets) et prise de décision (actions adaptées).
- Exemple : Accident de la Tesla Model S en 2016 (confusion entre une remorque blanche et le ciel).
- Objectif : Développer des outils performants pour une reconnaissance fiable, essentielle pour la sécurité routière.

# Présentation du jeu de données



## Préparation des données

```
datagen =  
ImageDataGenerator(  
    rotation_range=30,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)
```

```

for image_path in image_paths:
    image =
cv2.imread(image_path)
    if image is None:
        print(f"Erreur de lecture
de l'image: {image_path}")
        continue

    image = cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)
    image = cv2.resize(image,
(32, 32)) # Redimensionner
    image =
image.astype('float32') / 255.0 #
Normaliser
    images.append(image)

```

```
Forme de X (données images) : (39209, 32, 32, 3)  
Exemple de labels encodés : [[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
Forme de X (données images) : (39209, 32, 32, 3)  
Exemple de labels encodés : [[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.  
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

```
gray(images)
```

# Etalonnage du modèle

```
def create_model(optimizer):  
    model = models.Sequential()  
    model.add(layers.Input(shape=(32, 32, 3))) # Entrée 32x32 avec 3 canaux (RGB)  
    model.add(layers.Conv2D(32, (3, 3), activation='relu'))  
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))  
    model.add(layers.Conv2D(64, (3, 3), activation='relu'))  
    model.add(layers.Flatten())  
    model.add(layers.Dense(64, activation='relu'))  
    model.add(layers.Dense(43, activation='softmax')) # 43 classes  
    model.compile(optimizer=optimizer,  
                  loss='sparse_categorical_crossentropy', # Pour la classification multi-classes  
                  metrics=['accuracy'])  
  
    return model  
  
param_grid = {  
    'batch_size': [32, 64, 256],  
    'epochs': [10, 20],  
    'optimizer': ['adam', 'sgd'],  
    'learning_rate': [0.001, 0.01]  
}
```

batch_size	epochs	optimizer	learning_rate	avg_val_score
256	20	adam	0.001	0.9643893639246622

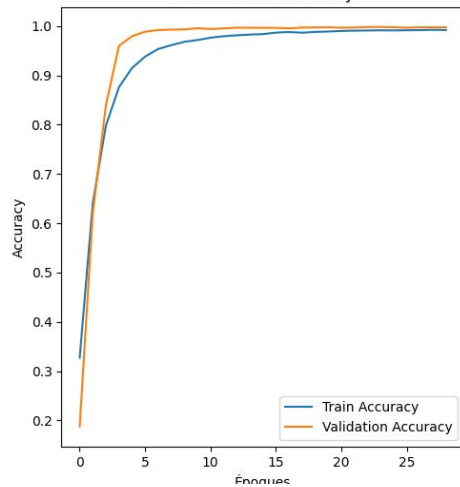
# Modèle de classification

Model: "sequential\_3"

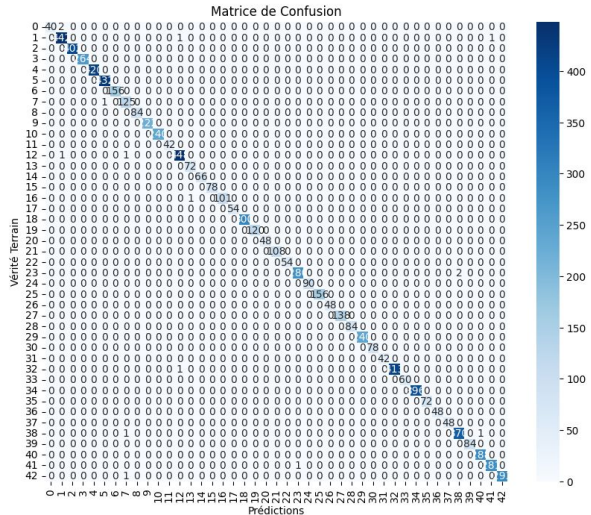
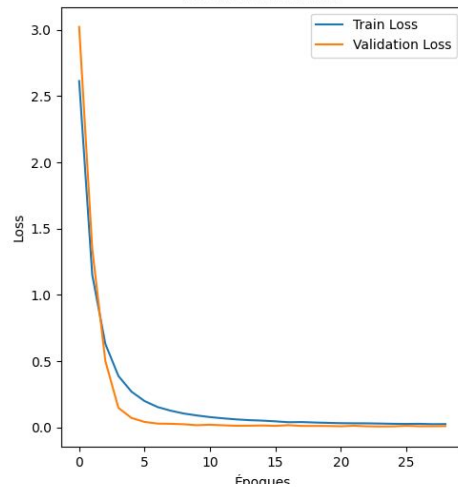
Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 30, 30, 32)	896
batch_normalization_15 (BatchNormalization)	(None, 30, 30, 32)	128
max_pooling2d_9 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_15 (Dropout)	(None, 15, 15, 32)	0
conv2d_13 (Conv2D)	(None, 13, 13, 64)	18,496
batch_normalization_16 (BatchNormalization)	(None, 13, 13, 64)	256
max_pooling2d_10 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_16 (Dropout)	(None, 6, 6, 64)	0
conv2d_14 (Conv2D)	(None, 4, 4, 128)	73,856
batch_normalization_17 (BatchNormalization)	(None, 4, 4, 128)	512
max_pooling2d_11 (MaxPooling2D)	(None, 2, 2, 128)	0
dropout_17 (Dropout)	(None, 2, 2, 128)	0
conv2d_15 (Conv2D)	(None, 2, 2, 256)	295,168
batch_normalization_18 (BatchNormalization)	(None, 2, 2, 256)	1,024
dropout_18 (Dropout)	(None, 2, 2, 256)	0
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 256)	0
dense_6 (Dense)	(None, 256)	65,792
batch_normalization_19 (BatchNormalization)	(None, 256)	1,024
dropout_19 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 43)	11,051

Total params: 1,491,567 (1.35 MB)

Évolution de l'accuracy



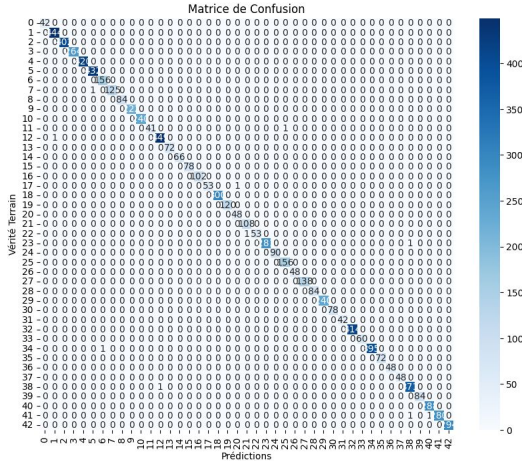
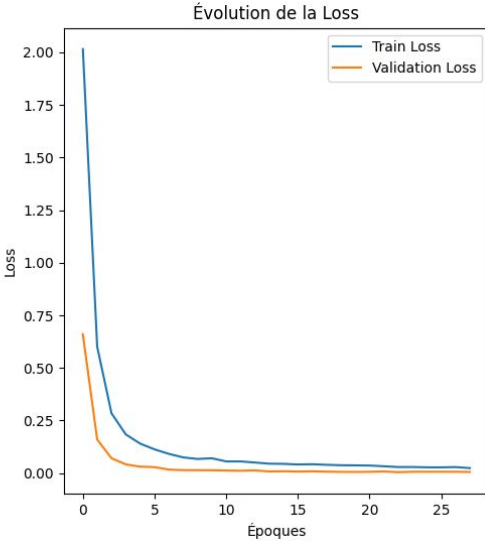
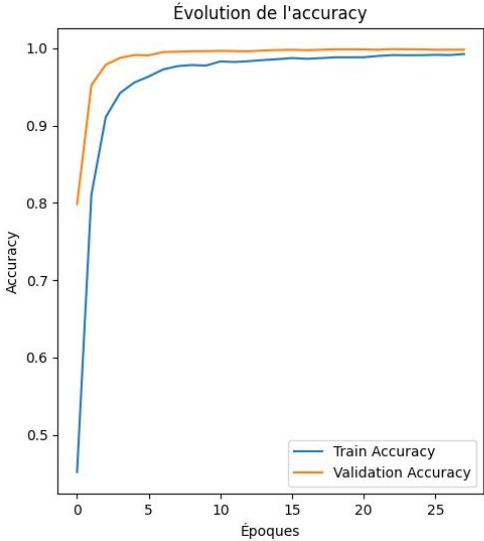
Évolution de la Loss



# Optimisation du modèle

Model: "sequential\_16"

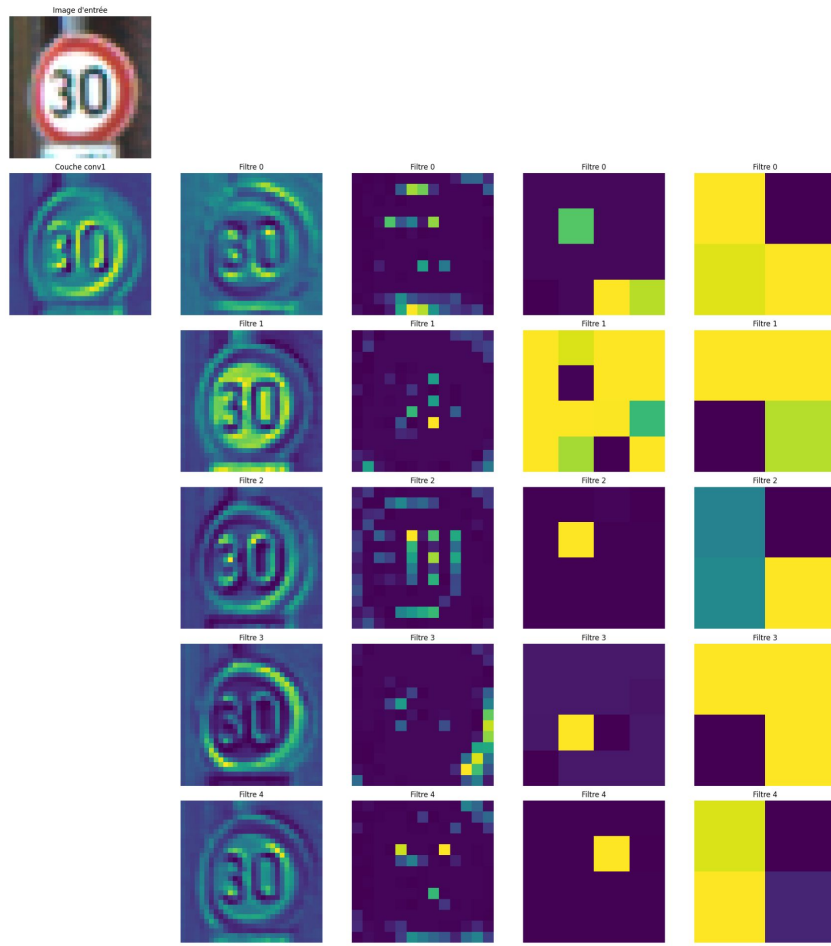
Layer (type)	Output Shape	Param #
conv2d_64 (Conv2D)	(None, 30, 30, 32)	896
batch_normalization_80 (BatchNormalization)	(None, 30, 30, 32)	128
max_pooling2d_48 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_80 (Dropout)	(None, 15, 15, 32)	0
conv2d_65 (Conv2D)	(None, 13, 13, 64)	18,496
batch_normalization_81 (BatchNormalization)	(None, 13, 13, 64)	256
max_pooling2d_49 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_81 (Dropout)	(None, 6, 6, 64)	0
conv2d_66 (Conv2D)	(None, 4, 4, 128)	73,856
batch_normalization_82 (BatchNormalization)	(None, 4, 4, 128)	512
max_pooling2d_50 (MaxPooling2D)	(None, 2, 2, 128)	0
dropout_82 (Dropout)	(None, 2, 2, 128)	0
conv2d_67 (Conv2D)	(None, 2, 2, 256)	295,168
batch_normalization_83 (BatchNormalization)	(None, 2, 2, 256)	1,024
dropout_83 (Dropout)	(None, 2, 2, 256)	0
global_average_pooling2d_16 (GlobalAveragePooling2D)	(None, 256)	0
dense_32 (Dense)	(None, 256)	65,792
batch_normalization_84 (BatchNormalization)	(None, 256)	1,024
dropout_84 (Dropout)	(None, 256)	0
dense_33 (Dense)	(None, 43)	11,051
Total params: 1,401,667 (5.35 MB)		
Trainable params: 466,731 (1.78 MB)		





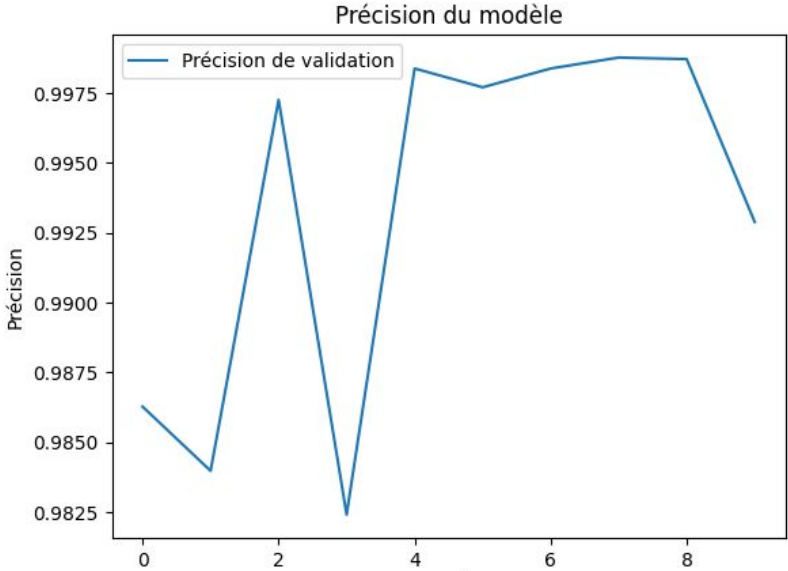
# Test et interprétation du modèle

## Modèle Swish

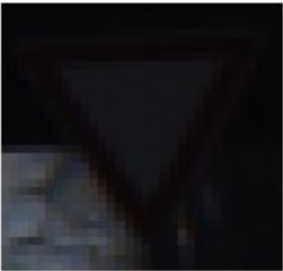




# Modèle binaire de classification de panneau



Vérité : Panneau    Vérité : Panneau    Vérité : Panneau    Vérité : Non-Panneau    Vérité : Non-Panneau  
Prédiction : Panneau    Prédiction : Panneau    Prédiction : Panneau    Prédiction : Non-Panneau    Prédiction : Non-Panneau



# Conclusion

## **Performances du modèle :**

- Précision de validation : 99.75%.
- Taux d'erreur faible, mais non nul (erreurs rares mais possibles).

## **Enjeux de la conduite autonome :**

- Chaque erreur peut avoir des conséquences graves (exemple : accident de la Tesla Model S en 2016).
- Importance de la perception et de la robustesse des systèmes.

## **Défis à relever :**

- Conditions environnementales difficiles (pluie, brouillard, neige).
- Variations de signalisation entre pays.
- Attaques adversariales (stickers, altérations visuelles).
- Nécessité de combiner vision par ordinateur avec LIDAR et radar.

## **Perspectives :**

- Améliorer la fiabilité des modèles pour atteindre une précision absolue.
- Contribuer à une conduite autonome plus sûre et réduire les accidents.

Modèle performant, mais des progrès restent nécessaires pour garantir une sécurité totale dans des conditions réelles.