

# Classification d'images JV

par Ludovic Sanne

# Intro

Le but de cette classification d'images est de reconnaître 5 personnages de jeux vidéos.



Yoshi



Daisy



Harmonie



Tiny Kong



Waluigi

# définitions

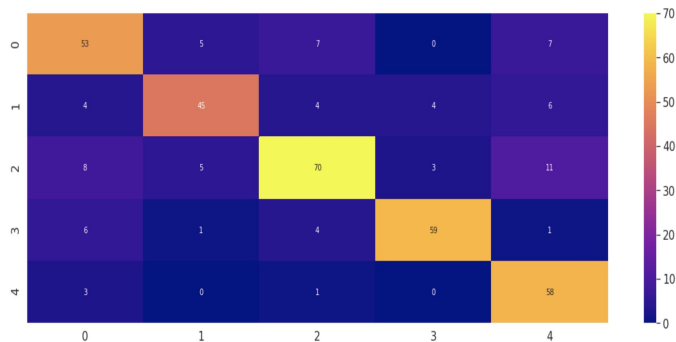
**Preprocessing (en informatique):** processus de préparation d'un ensemble de données afin de développer un modèle mathématique ou informatique.

**Matrice de confusion** : matrice qui mesure la qualité d'un système de classification. Chaque ligne correspond à une classe réelle, chaque colonne correspond à une classe estimée. Un de ses intérêts est qu'elle montre rapidement si un système de classification parvient à classifier correctement.

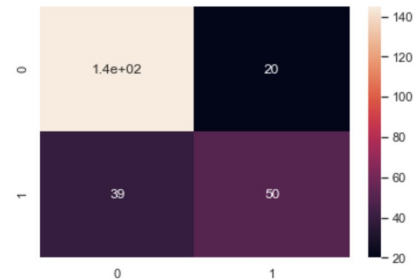
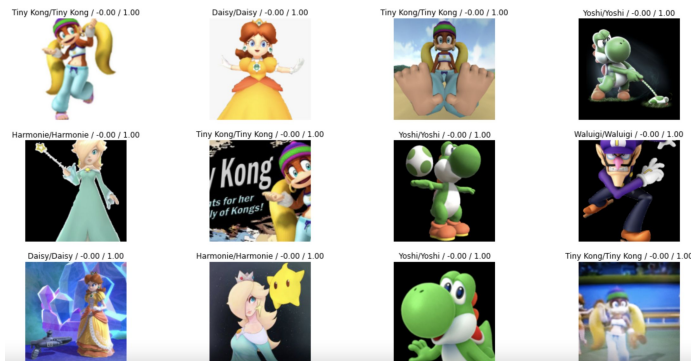
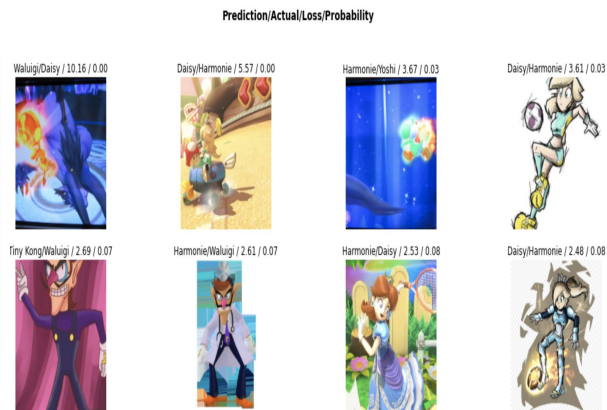
**Accuracy (exactitude):** Combinaison de 2 types d'erreur observationnelle (aléatoire et systématique), donc une grande précision exige à la fois une grande précision et une grande vérité. C'est aussi la description des erreurs systématiques, mesure des biais statistiques; une faible précision fait la différence entre un résultat et une « vraie » valeur.

**Gradient** : vecteur représentant la variation d'une fonction par rapport à la variation de ses différents paramètres .(Ex : le gradient d'une fonction  $f$  en un point  $M$  correspond au vecteur dont les composantes sont les dérivées partielles de  $f$  calculées au point  $M$ ).

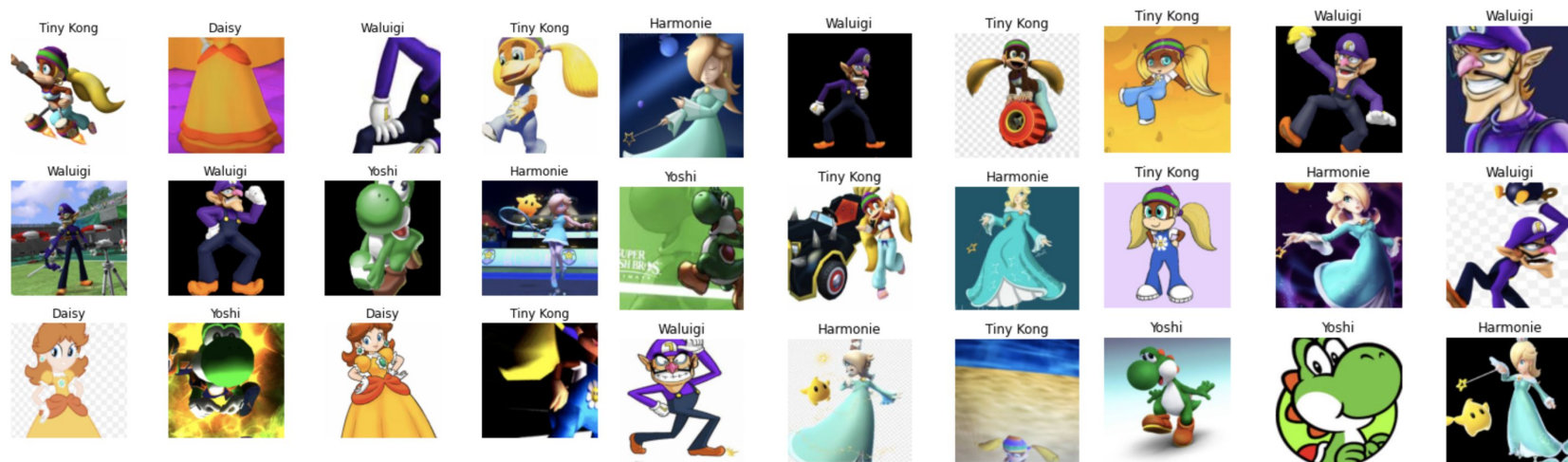
# Exemples de Matrices de confusion



	Daisy	Harmonie	Tiny Kong	Waluigi	Yoshi
Daisy	28	4	0	1	0
Harmonie	1	21	1	2	0
Tiny Kong	0	0	48	2	0
Waluigi	0	1	2	25	0
Yoshi	0	1	0	0	31

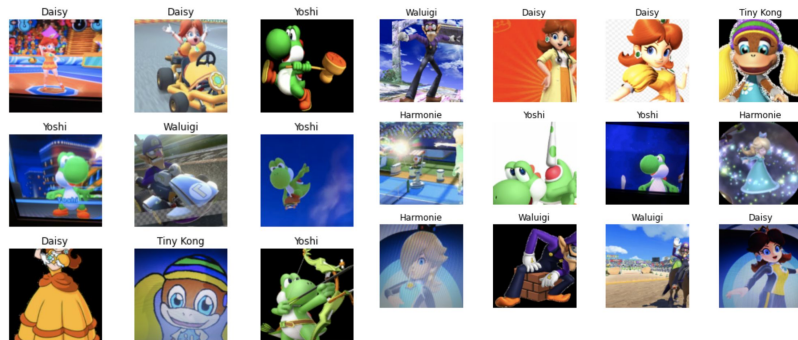


# Classification des images



Nombre d'images total : **2695**

Nombre de classes d'images : **5**



# Preprocessing

Création de 2 datasets : Train( regroupant 80 % des données) et Test (regroupant 20 % des données).

datasplit en x\_train ,y\_train, x\_test, et y\_test (pour les modèles prédictifs)

Nombre d'images dans le dataset Train : **1885**

Nombre d'images dans le dataset Test : **810**

# Résumé sous forme de tableau

Personnages	Nombre d'images	NB images (Train)	NB images (Test)
Yoshi	614	429	185
Daisy	633	443	190
Harmonie	533	373	160
Tiny Kong	416	291	125
Waluigi	499	349	150
total	2695	1885	810

# Modèles utilisés

Modèles: ResNet50, Vgg16, Vgg19, ResNet34

metric utilisé (valeur utilisée): Accuracy

	Resnet50	Vgg16	Vgg19	Resnet34
train	<b>0.9626666903495789</b>	0.8575000166893005	0.8374999761581421	0.9653333425521851
test	<b>0.95652174949646</b>	0.8349999785423279	0.800000011920929	0.9316770434379578

D'après ce tableau, le meilleur modèle, pour prédire les personnages , est le modèle **Resnet 50**



# Quelques définitions

**Resnet** : réseau neuronal artificiel (ANN) d'un genre s'appuyant sur des constructions connues à partir de cellules pyramidales dans le cortex cérébral. C'est aussi une forme d'«architecture exotique » s'appuyant sur des modules de micro-architecture (aussi appelés « architectures réseau en réseau »: ensemble des « blocs de construction » utilisés pour construire le réseau). On distingue plusieurs modèles en fonction du nombre de couches. (les plus connus sont : **Resnet34**, et **Resnet50**).

**VGG (Visual Geometry Group)** : réseau neuronal convolutionnel caractérisé par sa simplicité, en utilisant seulement 3×3 couches convolutionnelles empilées les unes sur les autres en profondeur croissante. La réduction de la taille du volume est gérée par une mise en commun maximale. 2 modèles sont connus: les modèles **Vgg16** et **Vgg19**. On les utilise dans de nombreux problèmes de classification d'image d'apprentissage profond.

# Resnet34 et Resnet50

Chaque bloc ResNet a 2 couches de profondeur (utilisées dans les petits réseaux comme ResNet 18, 34) ou trois couches de profondeur (ResNet 50, 101, 152).

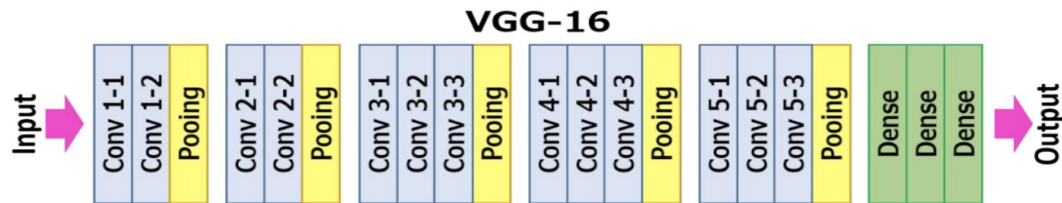
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Resnet 34 : Réseau Resnet avec 34 couches.

Resnet 50 : Chaque bloc à 2 couches est remplacé dans le filet à 34 couches par ce bloc goulot d'étranglement à 3 couches, ce qui donne un ResNet à 50 couches (voir le tableau ci-dessus).

# Vgg16 et Vgg19:

**Vgg16:** Réseau neuronal convolutionnel composé de 16 couches de profondeur (13 couches de convolution et 3 fully-connected). Il prend en entrée une image en couleurs de taille  $224 \times 224$  px et la classe dans une des 1000 classes.



**Vgg19** : Même chose que le Vgg16, mais avec 19 couches.



# VGG vs Resnet

	VGG	Resnet
Avantages	<ul style="list-style-type: none"><li>- Amélioration massive de la précision et de la vitesse (grâce à l'amélioration de la profondeur du modèle et l'introduction de modèles préformés).</li><li>- Augmentation de la non-linéarité toujours positif (grâce à l'augmentation du nombre de couches)</li></ul>	<ul style="list-style-type: none"><li>- L'architecture ResNet n'a pas besoin de tirer tous les neurones à chaque couche (réduisant considérablement le temps d'entraînement et améliorant la précision).</li><li>- La complexité d'un réseau VGG identique a causé le problème de dégradation qui a été résolu par l'apprentissage résiduel.</li></ul>
Inconvénients	<ul style="list-style-type: none"><li>- problème de gradient de disparition (rencontré quand lorsque plus de couches utilisant certaines fonctions d'activation sont ajoutées aux réseaux neuronaux, les gradients de la fonction de perte approchent de zéro, rendant le réseau difficile à former).</li><li>- Modèle très lent à entraîner</li></ul>	<ul style="list-style-type: none"><li>- complexité accrue de l'architecture</li><li>- Ajout de connexions de niveau pour lesquelles la dimensionnalité entre les différentes couches est prise en compte</li></ul>