

Projet VigneNette

Déroulement proposé

Problématique

Comment surveiller et détecter le plus tôt possible l'apparition de maladies sur les vignes ?

Problème actuel et concret

L'IA au service de la santé et de l'environnement



Comment lutter contre les maladies du bois ?

Les maladies du bois sont difficiles à prévenir et à éradiquer...

➔ En savoir plus

Source : https://www.agro.basf.fr/fr/cultures/vigne/maladies_de_la_vigne/maladies_du_bois/

Démarche pédagogique : projet

Compétences et connaissances

Innover

I6 : Évaluer une solution.

(Mesure et tests des performances de tout ou partie de la solution innovante, amélioration continue)

Analyser

A4 : Analyser le traitement de l'information.

(Algorithme, programme, notions sur **l'intelligence artificielle**)

Expérimenter / Simuler

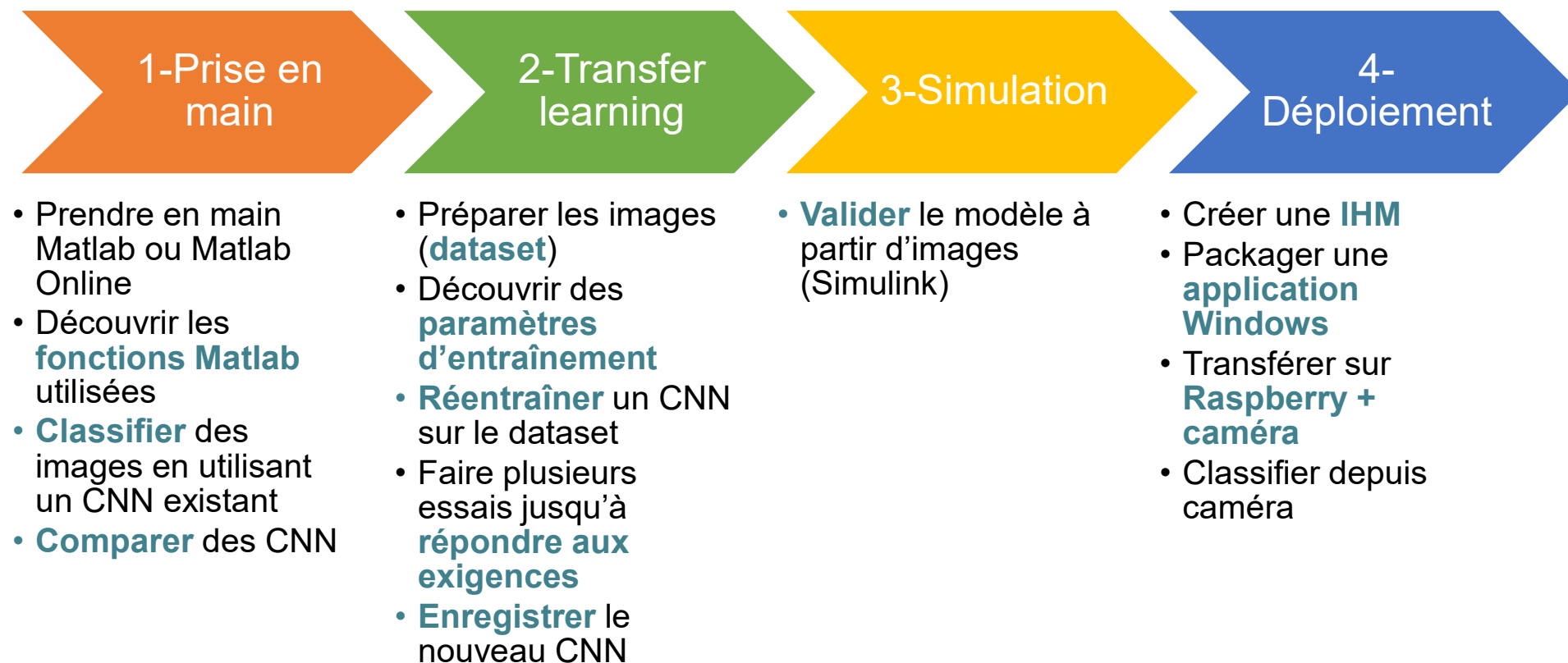
E7 : Modifier les paramètres influents et le programme de commande en vue d'optimiser les performances d'un produit

(Processus itératif d'amélioration des performances)

E9 : Valider un modèle numérique de l'objet simulé.

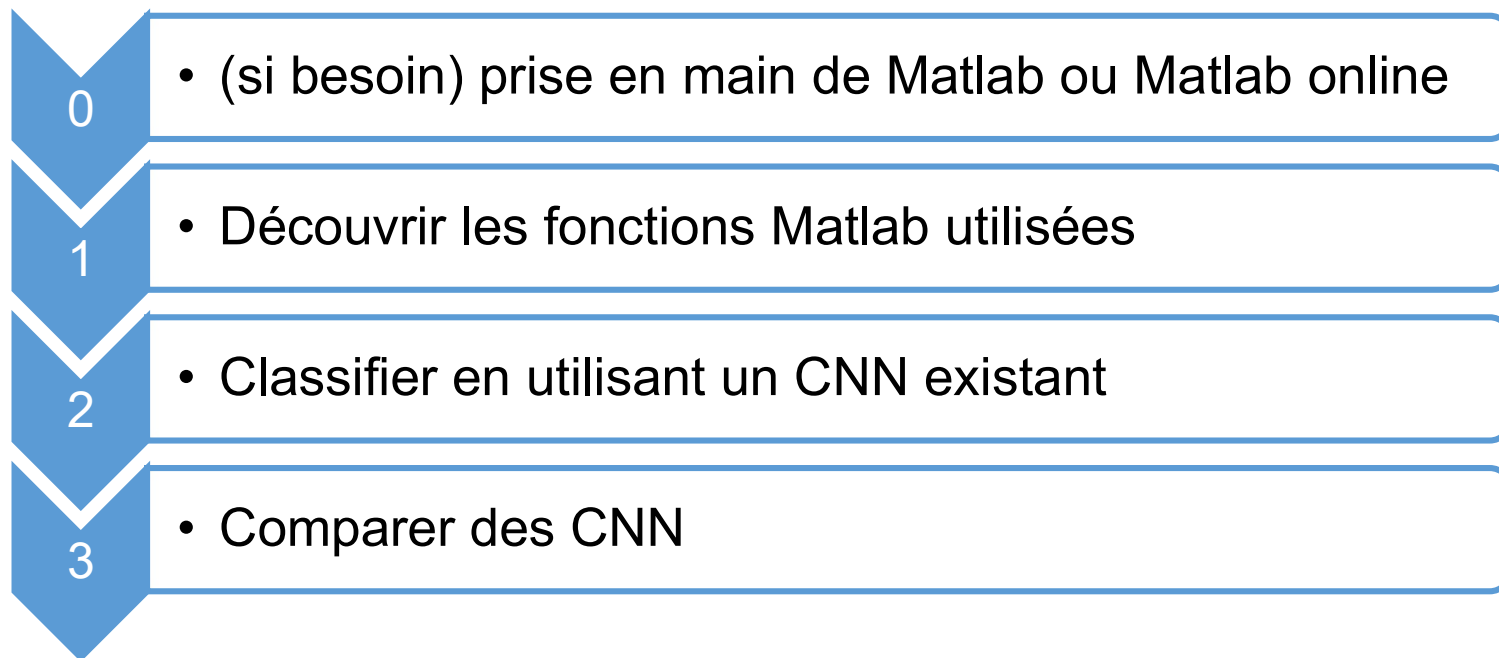
(Écarts entre les performances simulées et mesurées, limites de validité d'un modèle)

Les étapes de réalisation du projet



Les étapes du projet

Prise en main



Les étapes du projet

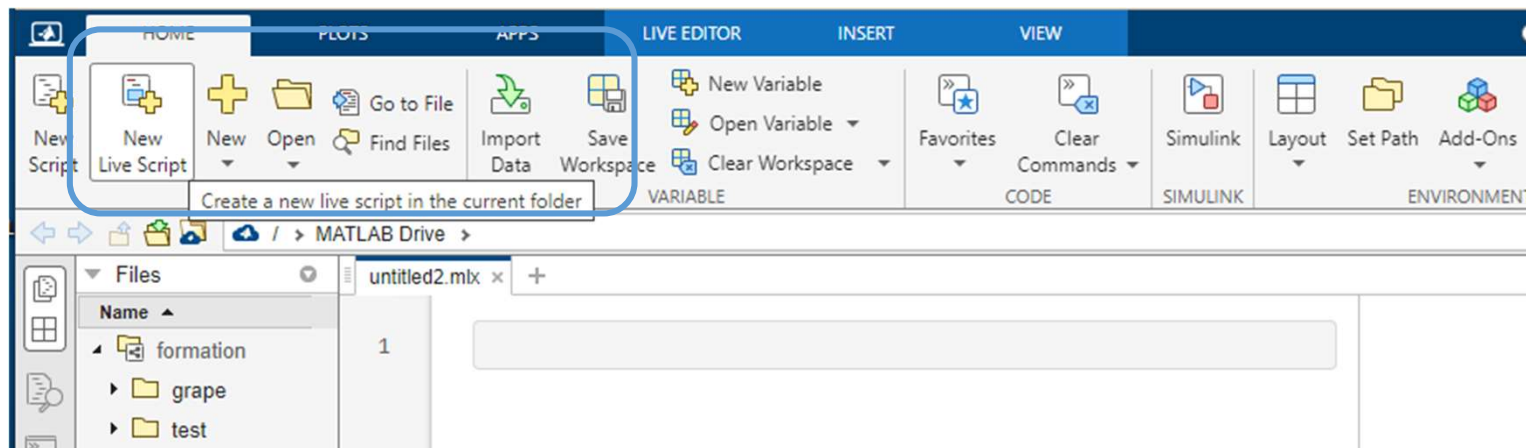
Prise en main



0

- Prise en main de Matlab Online

Créer un Live Script



Les étapes du projet

The screenshot shows the MATLAB Live Editor interface. The top ribbon includes tabs for HOME, PLOTS, APPS, LIVE EDITOR (highlighted in yellow), INSERT, and VIEW. The LIVE EDITOR tab contains sub-panels for Text, Code, Control, Task, and ANALYZE. The Code panel shows a script with two sections. The first section (lines 1-3) contains the code `a=23`, `b=42;`, and `a+b`. The second section (lines 4-5) contains `net = alexnet`. The right side of the interface shows the execution results for the first section: `a = 23` and `ans = 65`. The bottom left shows the Workspace pane with variables `a` (23), `ans` (65), and `b` (42).

Arborescence accès aux fichiers

Exécution de sections ou du script entier

Le script peut être découpé en plusieurs sections qui peuvent être exécutées indépendamment

Workspace = objets chargés en mémoire accessibles depuis les scripts

Le ';' annule l'affichage du résultat de l'instruction

Zone du résultat d'exécution des scripts (ici seule la première section a été exécutée)

Name	Value
a	23
ans	65
b	42

Les étapes du projet

Prise en main



The screenshot displays the MATLAB environment during the 'Prise en main' step. The script editor contains the following code:

```
1 a=23
2 b=42;
3 a+b
4
5 net = alexnet
```

The Workspace table shows the following variables:

Name	Value	Size	Class
a	23	1×1	double
ans	65	1×1	double
b	42	1×1	double
net	1×1 Series...	1×1	SeriesNetw...

The Command Window shows the output of the script execution:

```
a = 23
ans = 65
net =
  SeriesNetwork with properties:
    Layers: [25×1 nnet.cnn.layer.Layer]
    InputNames: {'data'}
    OutputNames: {'output'}
```

Blue arrows indicate the execution flow: one arrow points from the 'net = alexnet' line in the script to the workspace table, and another points from the 'net' variable in the workspace to the Command Window output. A text label 'Ici la deuxième section vient d'être exécutée' (Here the second section has just been executed) is positioned near the workspace table.

Les étapes du projet

Prise en main



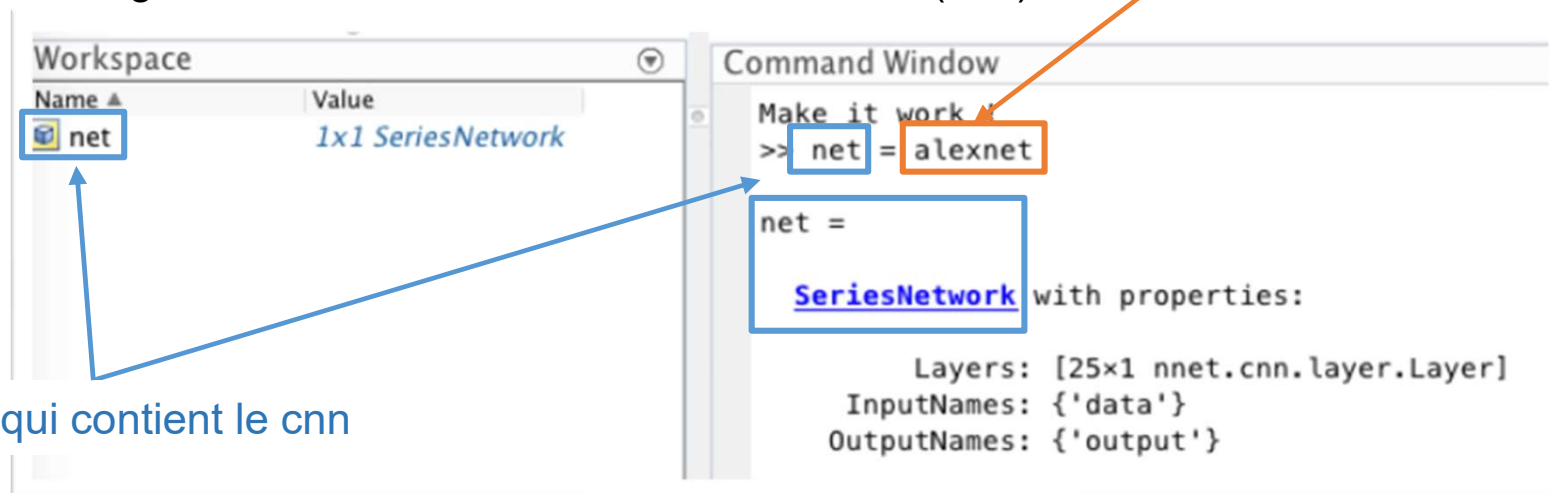
1

- Découvrir les fonctions Matlab utilisées

Charger un réseau de neurones à convolution (cnn)

Nom du cnn préexistant dans Matlab

Variable qui contient le cnn



Les étapes du projet

Prise en main



1

- Découvrir les fonctions Matlab utilisées

imread (Read image from graphics file) . Ex :

```
1 A = imread('ngc6543a.jpg');
```

<https://fr.mathworks.com/help/matlab/ref/imread.html>

imresize (Resize image). Ex :

```
1 B = imresize(A,[numrows numcols])
```

<https://fr.mathworks.com/help/matlab/ref/imresize.html>

classify (Classify data using trained deep learning neural network)

```
1 labels = classify(net,images)
```

<https://fr.mathworks.com/help/deeplearning/ref/seriesnetwork.classify.html>

Les étapes du projet

Prise en main

Prise en main

Transfer learning

Simulation

Déploiement

1

- Découvrir les fonctions Matlab utilisées

figure (Create Figure Window)

<https://fr.mathworks.com/help/matlab/ref/figure.html>

```
1 figure; % ouvre une fenêtre d'affichage (fenêtre d'affichage courante)
2 figure(1); %ouvre une fenêtre numérotée
```

imshow (Display Image)

(on pourra utiliser le paramètre « **InitialMagnification** »

<https://fr.mathworks.com/help/matlab/ref/imshow.html>

```
1 imshow(monImage);
2 imshow(monImage, 'InitialMagnification', 'fit');
```

title (Add title)

<https://fr.mathworks.com/help/matlab/ref/title.html>

```
1 % plusieurs façons d'écrire le titre de la figure
2 title('Le titre de la figure');
3 title(char(leTitre));
4 title(string(leTitre));
5
```

Les étapes du projet

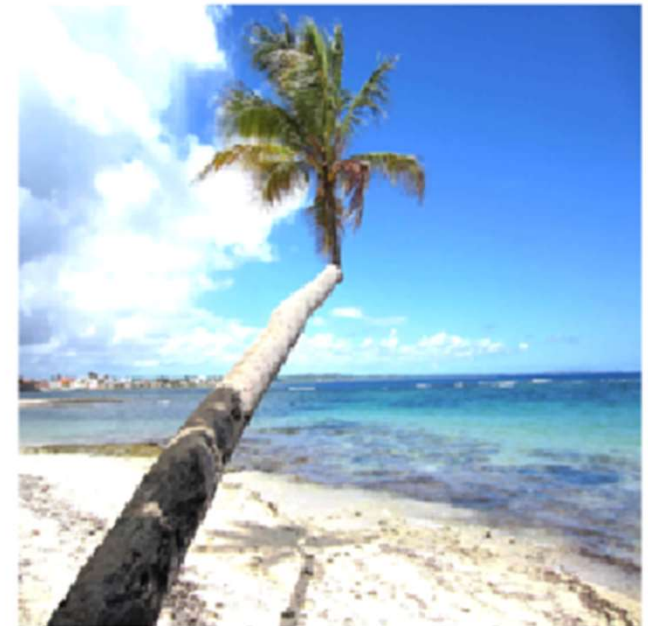
Prise en main



2

- Classifier en utilisant un CNN existant
- Obtenir le label d'une image

seashore



Les étapes du projet

Prise en main



3

- Comparer des CNN

- Classification avancée
 - Obtenir les probabilités des 5 premiers labels



```
imshow(Aresized,'InitialMagnification','fit');
title(string(labels))

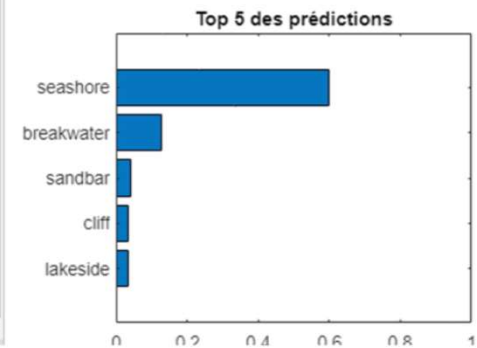
Classification avancée d'une image

leslabels = net.Layers(end).ClassNames;
% récupère tous les labels avec le score attribué dans un tableau
[label, scores] = classify(net, Aresized);
% trie le tableau par ordre décroissant et récupère les index
[~,idx] = sort(scores,'descend');

% prend les 5 premiers index du tableau
idx = idx(5:-1:1);

% récupère les classes et les scores pour ces indices
classNamesTop = net.Layers(end).ClassNames(idx);
scoresTop = scores(idx);

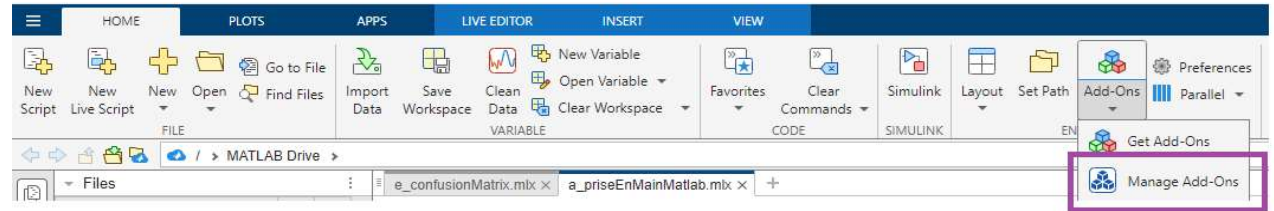
% affiche une nouvelle figure avec un graphique en barres
figure(2)
barh(scoresTop)
xlim([0 1])
title('Top 5 des prédictions')
xlabel('Probabilité')
yticklabels(classNamesTop)
```



Les étapes du projet

Prise en main

Prise en main



3

- Comparer des CNN
- Matlab Home > Add-Ons > Manage Add-Ons
 - Liste des CNN disponibles

	Deep Learning Toolbox Model for DarkNet-53 Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for DenseNet-201 Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for EfficientNet-b0 Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for GoogLeNet Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for Inception-ResNet-v2 Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for Inception-v3 Network version 23.1.0		MathWorks	7 April 2023
	Deep Learning Toolbox Model for MobileNet-v2 Network version 23.1.0		MathWorks	7 April 2023

Les étapes du projet

Prise en main



3

- Comparer des CNN
- Trouver le nom du CNN
 - View in Add-Ons explorer
 - Rubrique « Usage example »

Overview

Reviews (9)

Discussions (0)

MobileNetv2 is a pretrained model that has been trained on a subset of the ImageNet dataset. The model is trained on more than a million images and can classify images into 1000 classes (e.g. keyboard, mouse, pencil, and many animals).

Opening the mobilenetv2.mlpkginstall file from your operating system or from within MATLAB will initiate the installation process for the release you have.

This mlpkginstall file is functional for R2019a and beyond.

Usage Example:

```
% Access the trained model
net = mobilenetv2();
```


Les étapes du projet

Prise en main



3

- Comparer des CNN



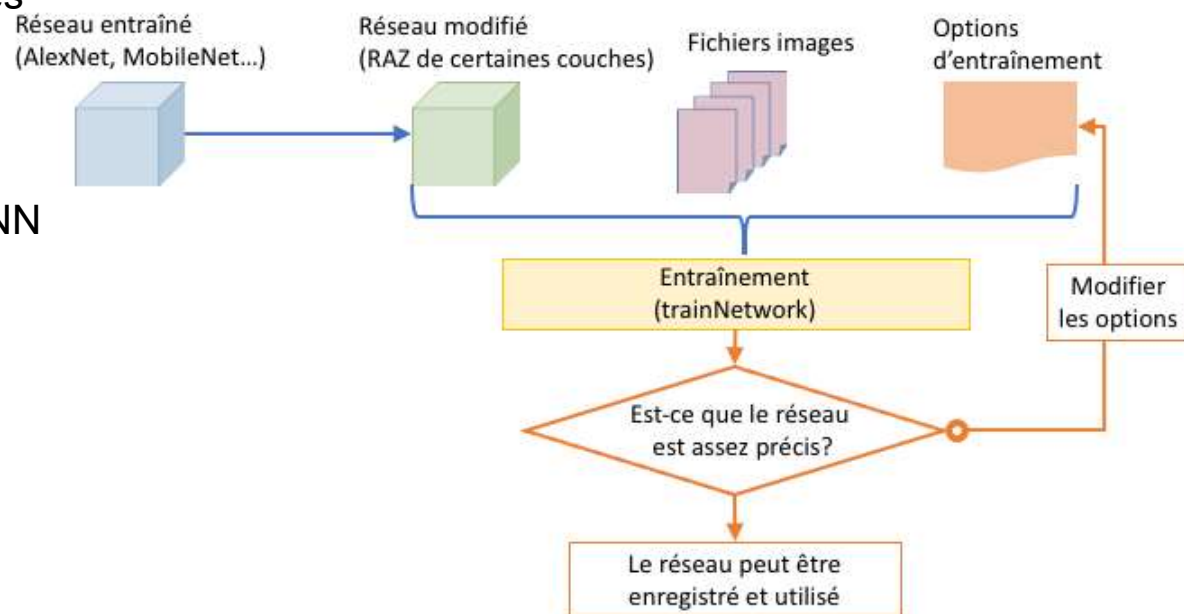
- Comparaison qualitative de plusieurs CNN
- Image au choix + une image de feuille
- Choisir le réseau qui sera ré-entraîné (ici GoogleNet)

1		carlin				
2		alexnet	squeezenet	googlenet	mobilenetv2	res-net18
3	score	99,96	99,98	98,91	78,22	98,61
4	temps	0.212721	0.477291	0.833171	2.913375	1.101561
5	Top1	pug	pug	pug	pug	pug
6	top2	Brabancon griffon	bull mastiff	Pekinese	Brabancon griffon	Norwegian elkhound
7	top3	French bulldog	French bulldog	Brabancon griffon	Pekinese	French bulldog
8	top4	Pekinese	Pekinese	bull mastiff	bull mastiff	Pekinese
9	top5	bull mastiff	Norwegian elkhound	Norwegian elkhound	shopping cart	bull mastiff
10						
11		feuille de bouleau				
12		alexnet	squeezenet	googlenet	mobilenetv2	res-net18
13	score	26,63	28,34	29,46	18,27	46,92
14	temps	0.234208	0.404576	0.2946	2.187315	1.095072
15	Top1	cabbage butterfly	acorn	buckeye	cucumber	head cabbage
16	top2	sulphur butterfly	cabbage butterfly	custard apple	pot	buckeye
17	top3	spaghetti squash	head cabbage	fig	buckeye	cucumber
18	top4	head cabbage	cauliflower	bell pepper	head cabbage	spaghetti squash
19	top5	cauliflower	walking stick	acorn	cauliflower	head cabbage

Les étapes du projet



- Préparer les images (**dataset**)
- Découvrir des **paramètres d'entraînement**
- **Réentraîner** un CNN sur le dataset
- Faire plusieurs essais jusqu'à **répondre aux exigences**
- **Enregistrer** le nouveau CNN



Les étapes du projet

Transfer learning



- Préparer les images (Dataset)

- Datastore
 - Objet Matlab pour lire ou écrire dans une grande quantité de données ou fichiers : pointe sur un répertoire contenant les données
 - Matlab charge les données en mémoire au fur et à mesure des opérations
- ImageDatastore
 - Datastore spécifique pour les images et le deep learning
 - Pointe sur un répertoire
 - Obtention automatique des **labels**
 - **AugmentedImageDatastore** : ajoute du pré-processing aux images avant le traitement par le modèle (changement de taille...)

Les étapes du projet

Transfer learning

Prise en main

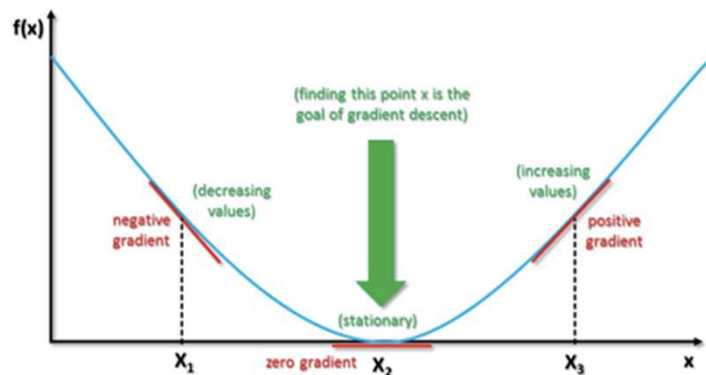
Transfer learning

Simulation

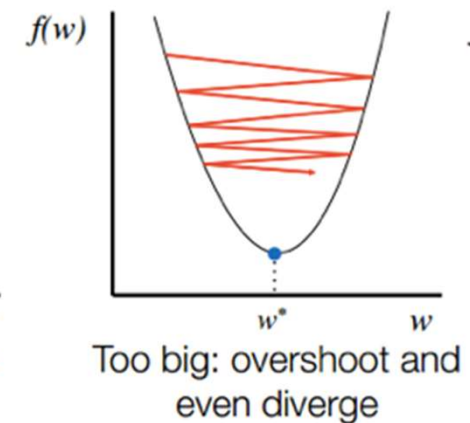
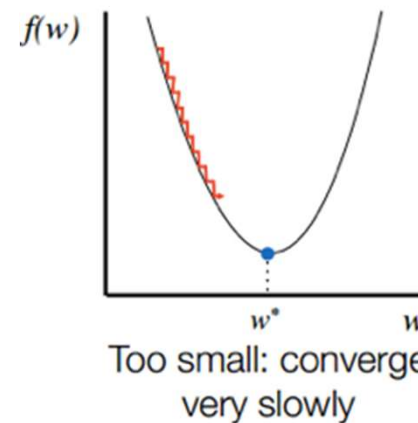
Déploiement

2

- Découvrir des paramètres d'entraînement



- Objectif : trouver le minimum de la fonction loss



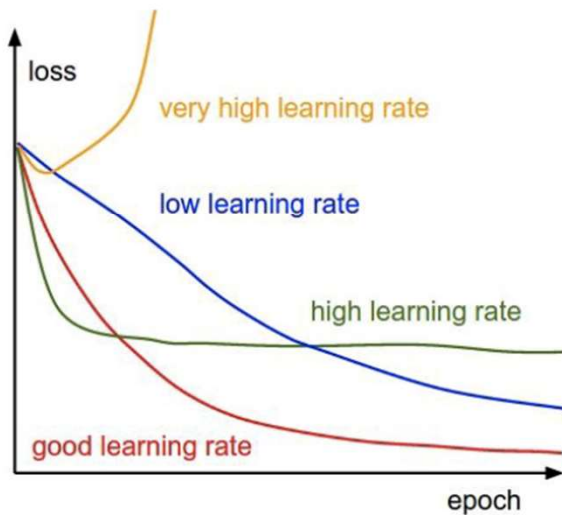
Les étapes du projet

Transfer learning



2

- Découvrir des paramètres d'entraînement



Descente de gradient

- Pendant l'entraînement, l'algorithme travaille sur des lots d'images (mini-batches)
- L'algorithme ajuste les paramètres après avoir comparé les **labels prédits** avec les **labels réels** sur le mini-batch actif
- Une époque = toutes les images ont été testées dans les mini-batches
- Calculer de préférence nombre d'images d'entraînement divisible par la taille de mini-batch

Les étapes du projet

Transfer learning



2

- Découvrir des paramètres d'entraînement

```
options = trainingOptions("sgdm", ...  
    LearnRateSchedule="piecewise", ...  
    LearnRateDropFactor=0.2, ...  
    LearnRateDropPeriod=5, ...  
    MaxEpochs=20, ...  
    MiniBatchSize=64, ...  
    Plots="training-progress")
```

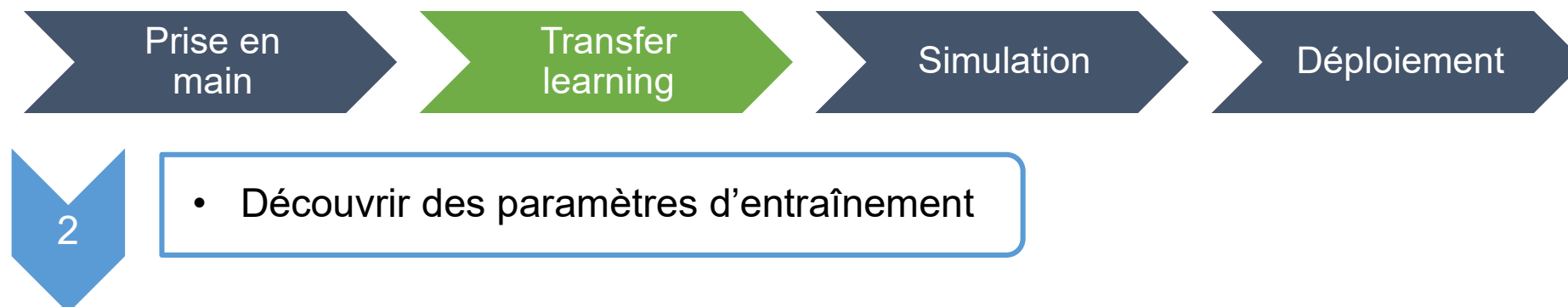
Algorithmes d'entraînement

Nombre d'images des mini-batches

Source : <https://fr.mathworks.com/help/deeplearning/ref/trainingoptions.html>

Les étapes du projet

Transfer learning



Field	Description
InitialLearningRate	Rythme d'apprentissage initial (par défaut 0,01 pour 'sgdm' et 0,001 pour 'adam')
LearnRateSchedule	Définit si le rythme d'apprentissage reste constant toutes les X époques
LearnRateDropPeriod	Nombre d'époques au bout desquelles le rythme d'apprentissage baissera
Shuffle	Option pour « mélanger » les images (par exemple pour chaque époque, les images seront redistribuées dans les mini-batches)

Source : <https://fr.mathworks.com/help/deeplearning/ref/trainingoptions.html>

Les étapes du projet

Transfer learning

Prise en main

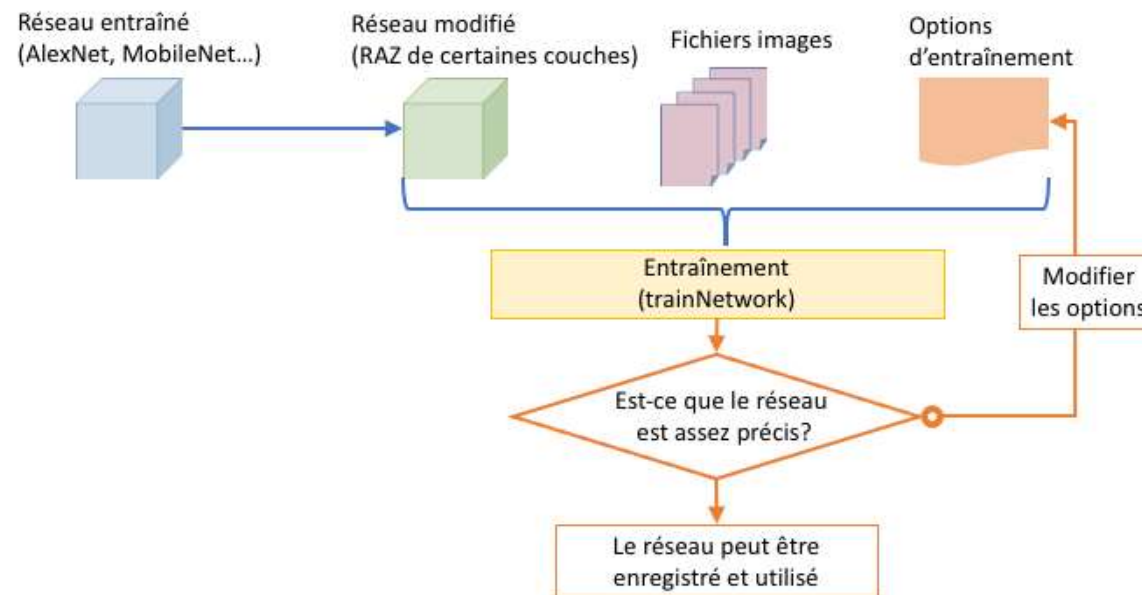
Transfer learning

Simulation

Déploiement

3

- Réentraîner un CNN sur le dataset



Les étapes du projet

Transfer learning

Prise en main

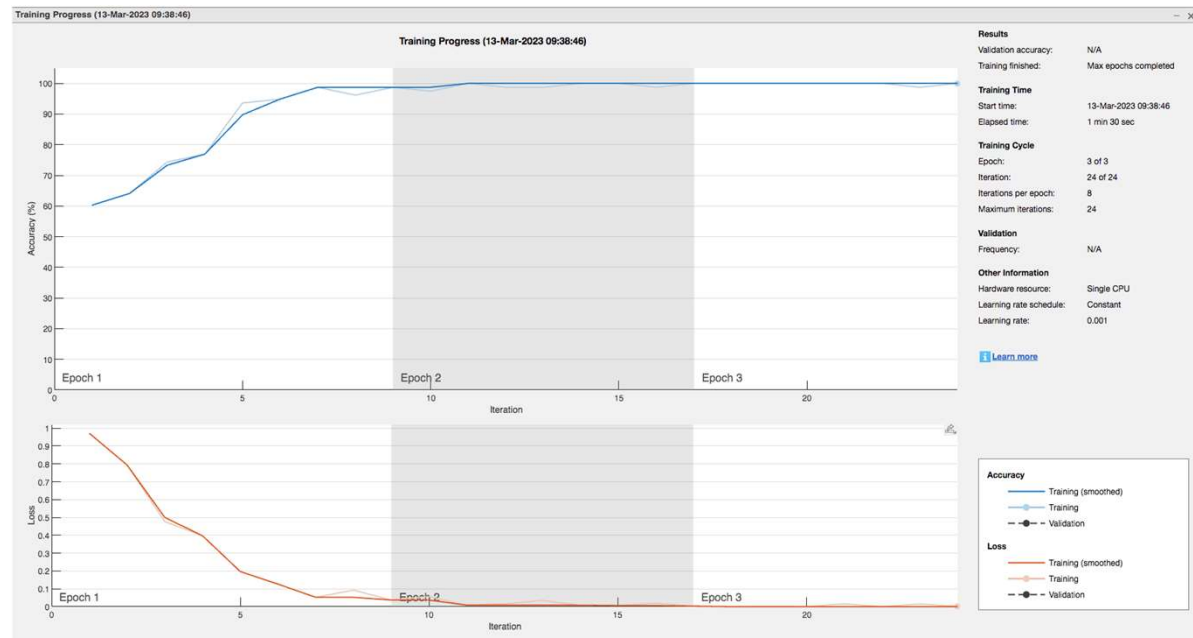
Transfer learning

Simulation

Déploiement

4

- Faire plusieurs essais jusqu'à répondre aux exigences



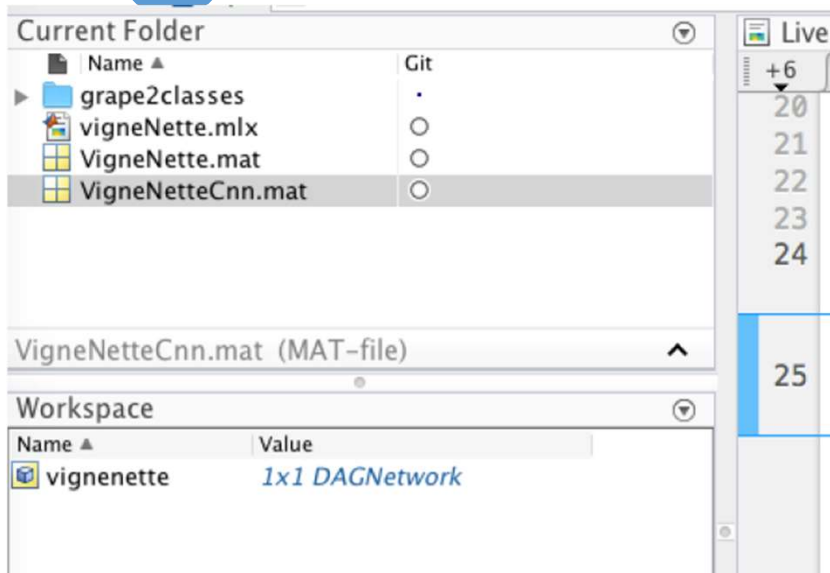
Les étapes du projet

Transfer learning



5

- Enregistrer le nouveau CNN



- Pour enregistrer le nouveau CNN
 - Supprimer toutes les variables du workspace SAUF le nouveau réseau de neurones
 - Enregistrer la variable de type DAG dans un nouveau fichier matlab .mat

Les étapes du projet

Transfer learning



5

- Enregistrer le nouveau CNN

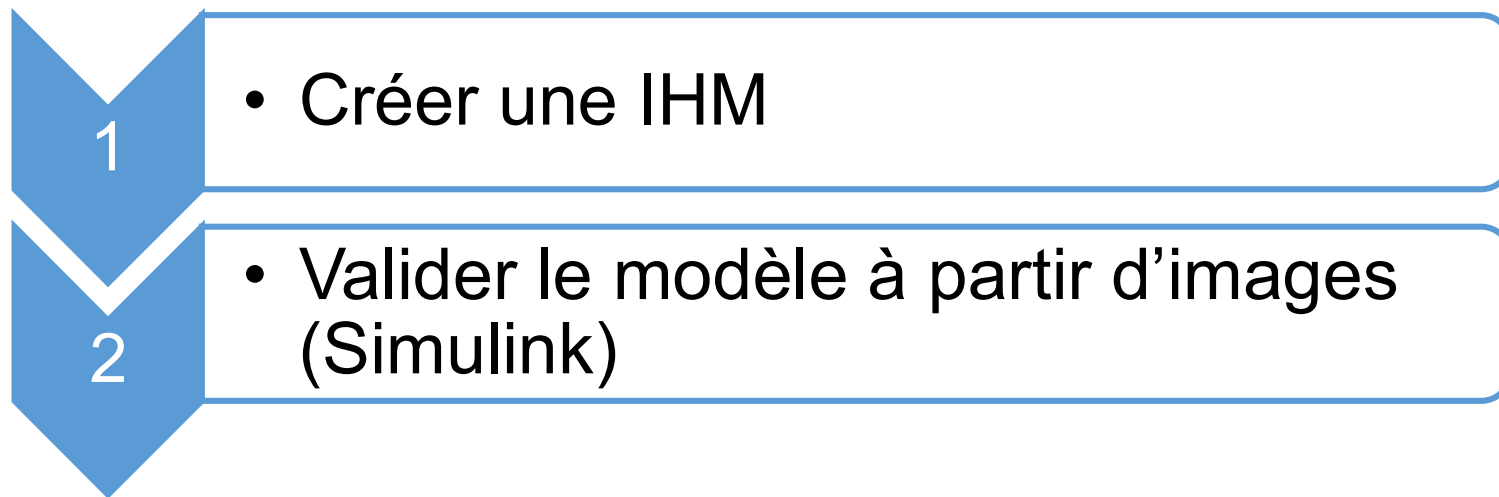
- Pour réutiliser le réseau entraîné
 - Charger le fichier Matlab contenant le CNN

% charge le **fichier Matlab** contenant le cnn obtenu par transfer learning
`load VigneNetteCnn.mat;`

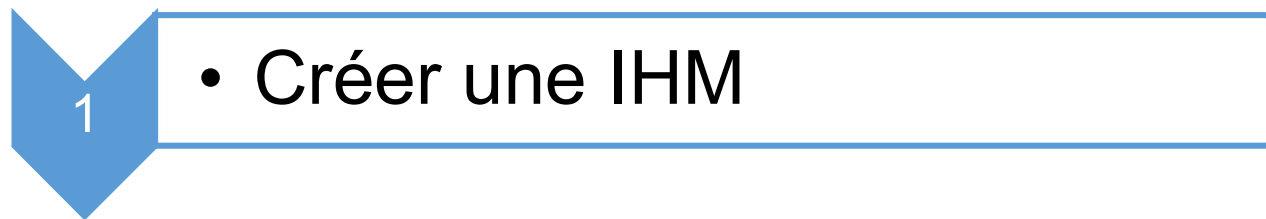
% charge le **cnn** dans une variable pour pouvoir l'utiliser (par exemple obtenir le label)
`net = vignenette;`

- On peut ensuite charger le CNN comme précédemment

Les étapes du projet

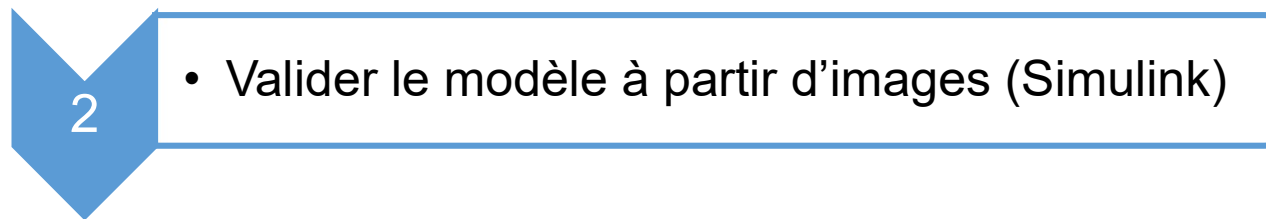


Les étapes du projet



[https://tonnerre.canoprof.fr/eleve/Ressources-SI/IA%20VigneNette%20\(D%C3%A9tection%20des%20maladies%20sur%20les%20feuilles%20de%20vigne%20-%20Deep%20Learning%20avec%20Matlab\)/activities/02-IHM-VigneNette.html](https://tonnerre.canoprof.fr/eleve/Ressources-SI/IA%20VigneNette%20(D%C3%A9tection%20des%20maladies%20sur%20les%20feuilles%20de%20vigne%20-%20Deep%20Learning%20avec%20Matlab)/activities/02-IHM-VigneNette.html)

Les étapes du projet



- Valider le modèle à partir d'images (Simulink)

Proposition de répartition - Semaine 1

3h

3h

étudiant1

- Analyser le besoin, analyser l'existant, analyser le cahier des charges
- Faire émerger des pistes de solutions / Brainstorming - Cartes heuristiques
- Définir le planning et répartir les tâches

étudiant2

étudiant3

- Prise en main des CNN sous Matlab / Matlab online

Proposition de répartition - Semaine 2



3h



3h

étudiant1	<ul style="list-style-type: none">• Étude qualitative des CNNs• Étude quantitative des CNNs	<ul style="list-style-type: none">• Synthèse des études, comparatif• Choix des 2 réseaux à entraîner
étudiant2	<ul style="list-style-type: none">• Étude qualitative des CNNs• Étude quantitative des CNNs	
étudiant3	<ul style="list-style-type: none">• Préparation carte Raspberry pour Matlab	<ul style="list-style-type: none">• Prise en main Simulink / carte Raspberry

Proposition de répartition - Semaine 3



3h



3h

étudiant1	<ul style="list-style-type: none">• Etude des paramètres d'entraînement• Synthèse et mise au point d'une stratégie d'entraînement• Test entraînement du CNN1	<ul style="list-style-type: none">• Entraînement du cnn 1, algorithme adam, avec différents paramètres• Modification d'un paramètre à la fois, sur petit nombre d'époques• Conclusions
étudiant2	<ul style="list-style-type: none">• Etude des paramètres d'entraînement• Synthèse et mise au point d'une stratégie d'entraînement• Test entraînement du CNN2	<ul style="list-style-type: none">• Entraînement du cnn 2, algorithme adam, avec différents paramètres• Modification d'un paramètre à la fois, sur petit nombre d'époques• Conclusions
étudiant3	<ul style="list-style-type: none">• Préparation du dataset• Préparation d'images complémentaires pour évaluer le modèle• Déploiement d'une application avec un réseau de neurones existant (exemple : Googlenet) sur la Raspberry (reconnaissance d'objets)	<ul style="list-style-type: none">• Suite du déploiement d'une application avec un réseau de neurones (exemple : Googlenet) sur la Raspberry (reconnaissance d'objets)

Proposition de répartition - Semaine 4



3h



3h

étudiant1	<ul style="list-style-type: none">• Entraînement du cnn 1, algorithme sgdm, avec différents paramètres• Ajustement des paramètres pour améliorer le cnn (tests sur images et matrice de confusion)• Conclusions	<ul style="list-style-type: none">• Synthèse des entraînements• Choix du réseau final• Entraînement du réseau avec les paramètres optimisés, sur un nombre d'époques plus élevés• Bilan : matrice de confusion, validation des performances (diagramme d'exigences)• Enregistrement du cnn VigneNet
étudiant2	<ul style="list-style-type: none">• Entraînement du cnn 2, algorithme sgdm, avec différents paramètres• Ajustement des paramètres pour améliorer le cnn (tests sur images et matrice de confusion)• Conclusions	
étudiant3	<ul style="list-style-type: none">• Déploiement d'une application avec un réseau de neurones (exemple : Googlenet) sur la Raspberry (reconnaissance d'objets) (suite)	<ul style="list-style-type: none">• Déploiement d'une application avec un réseau de neurones entraîné• Test et validation des performances (nombre d'images labellisées par minute...)

Proposition de répartition - Semaine 5



3h



3h

étudiant1	<ul style="list-style-type: none">• Validation du réseau sur des images variées• Mesure des écarts avec la précédente matrice de confusion• Reprise de l'entraînement si besoin• Enregistrement du réseau de neurones VigneNet (version définitive)	<ul style="list-style-type: none">• Application Windows : écriture du code
étudiant2	<ul style="list-style-type: none">• Ecriture d'un programme utilisant le réseau de neurones, affichage de la prédiction et de la probabilité	<ul style="list-style-type: none">• Application Windows : création de l'interface utilisateur
étudiant3	<ul style="list-style-type: none">• Intégration du cnn VigneNet à l'application Raspberry (image venant de la caméra / images fixes) avec le CNN VigneNet• Simulation (Simulink) et tests (images fixes / caméra : précision, temps de réponse)	<ul style="list-style-type: none">• Intégration de VigneNet sur l'application pour Raspberry (simulink)• Tests : précision, temps de réponse

Proposition de répartition - Semaine 6



3h



3h

étudiant1	<ul style="list-style-type: none">• Application Windows : suite	<ul style="list-style-type: none">• Finalisation de l'application Windows
étudiant2		
étudiant3	<ul style="list-style-type: none">• Déploiement sur Raspberry• Tests : temps de réponse, nb d'images par minute	<ul style="list-style-type: none">• Fin de l'intégration sur Raspberry• Mesure des écarts simulation / réel• Consignation des paramètres à utiliser (nb images par minute)

Proposition de répartition - Semaine 7

3h

3h

étudiant1	<ul style="list-style-type: none">• Ajout d'une communication MQTT (Thingspeak)	<ul style="list-style-type: none">• Fin alerte MQTT
étudiant2		
étudiant3	<ul style="list-style-type: none">• Création d'un dashboard Thingspeak• Ajout d'une communication MQTT (Thingspeak) pour alerter en cas de maladie	<ul style="list-style-type: none">• Fin alerte MQTT Thingspeak

Proposition de répartition - Semaine 8



3h



3h

étudiant1

- Packaging de l'application (installateur Windows)
- Tests unitaires (installation de l'application Windows, alertes)
- Validation des exigences
- Compte-rendu

étudiant2

étudiant3

- Synthèse
- Préparation du support de communication