Rapport de stage

Utilisation des CNN dans le contrôle qualité de tissages micrométriques en acier

Auteur :
Maureen BOUDART

Tuteurs :
Fabien PIERRE
Pierre-Frédéric VILLARD

16 mars-25 septembre 2020

1 Résumé

Le but est de réaliser une preuve de concept pour savoir s'il est possible de détecter des défauts dans les tissages micrométriques en acier. De bons résultats ont pu être obtenus en utilisant la classification automatique des images de tissages malgré la petite taille de la base de données. Un second axe a été exploré durant le stage, sur la classification de textures à différentes échelles mais dont la réalisation reste à finaliser.

2 Abstract

The aim is to realize a proof of concept to determine if it is possible to detect defects in micrometric steel meshes. Good results were obtained by using automatic classification of mesh images despite the small size of the database. A second axis has been explored during the internship, on the classification of textures at different scales but whose implementation remains to be finalized.

3 Remerciements

Je souhaite tout d'abord remercier Olivier Caspary, le directeur, de m'avoir accueillie à l'IUT de Saint-Dié-Des-Vosges pour mon stage de fin d'études.
Je souhaite remercier Fabien Pierre et Pierre-Frédéric Villard, mes tuteurs de stages pour leur accueil, leur encadrement, leur aide et leur temps qu'ils m'ont accordé durant ces 6 mois.
Je souhaite remercier Alain Mourot et Denis Antoine, ingénieurs chez Gantois pour la réalisation de la base de données.
Je souhaite remercier Charles Rocha d'avoir mis à ma disposition une machine en accès distant afin de pouvoir travailler avec une machine plus puissante dans le cadre de mon stage.

Table des matières

4 Introduction

4.1 Descriptif entreprise

Le stage s'est déroulé en télétravail pour l'institut universitaire de technologie (IUT) de Saint-Dié-Des-Vosges, composante de l'Université de Lorraine. Cet IUT compte 300 étudiants et propose trois types de formation :

— Génie électrique et informatique industrielle
— Informatique
— Métiers du multimédia et de l'internet

Les enseignants-chercheurs de ces formations exercent des activités de recherche au sein de différents laboratoires. :

— Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
— Centre de Recherche en Automatique de Nancy (CRAN)
— l'Institut Jean Lamour (IJL)
— Analyse et Traitement Informatique de la Langue Française (ATILF)

Les enseignants chercheurs de l'IUT de Saint-Dié-des-Vosges exerçant leurs recherches au sein du LORIA ont une thématique orienté "Image et Géométrie" dans laquelle s'inscrit le sujet de ce stage.

4.2 Contexte

Le sujet est un projet d'automatisation de contrôle qualité pour l'entreprise Gantois, partenaire de l'IUT.

Gantois est une industrie créée en 1894, spécialiste des métaux tissés et perforés . Elle est devenue Gantois Industries en 2011 filiale du groupe Drouault Industries. En architecture les tôles perforées servent en habillement de façade, à la réalisation de gardes-corps, de brises-soleils, d'aménagements urbains ou d'escaliers. Les tôles perforées ainsi que les toiles tissées ont également des applications industrielles. L'entreprise est certifiée ISO9001, certifiant un système de management dans la qualité de la conception, développement et fabrication de produits métalliques tissés, perforés et soudés. Elle est également certifiée ISO9100 certifiant le management de la qualité dans la fabrication de produits perforés, de toiles métalliques et de pièces finies à base de toiles métalliques destinées à des applications aéronautiques, spatiales et de défense permettant de livrer Airbus en rang 1.

Parmi les toiles tissées produites par Gantois, des tissages micrométriques en acier sont destinés à l'insonorisation de moteurs d'Airbus. La qualité étant une des valeurs de l'entreprise, celle-ci souhaite automatiser le processus de contrôle qualité de ses tissages. Dans un premiers temps l'objectif de ce sujet

est de réaliser une preuve de concept pour savoir s'il est possible de détecter des défauts sur les tissages micrométriques.

Ce système de détection de défaut reposera sur de la vision par ordinateur avec des objets non-géométriques mais texturés. Ces images s'opposent aux images représentant une forme comme un chat par exemple. Les images de grilles n'ont pas de contour mais représentent un motif qui se répète. Une approche pour détecter les défaut dans ces images consistera dans ce stage à considérer que les images de grilles sont des textures et que l'on utilisera des algorithmes de classification pour détecter les défauts.

Il n'y a pas de définition formelle de ce qu'est une texture. Il peut s'agir d'un motif qui peut être plus ou moins régulier ou aléatoire. Rupet Paget [9, chapitre 2, Rupet Paget] classe les textures en deux groupes, stochastique où le motif est aléatoire et régulier qui contient des motifs périodiques. Cependant une texture peut être également définie comme un mélange de d'éléments stochastiques et d'éléments structurés [9, chapitre 1].

Les images de tissages ne sont pas géométriques au sens où elles ne décrivent pas la forme d'un objet. Elles ont des motifs réguliers avec des effets de randomisations pouvant provenir légèrement de l'éclairage ou d'un éventuel bruit. Il nous semble donc raisonnable de considérer ces images comme de la texture et d'y appliquer des méthodes de l'état de l'art et de s'en inspirer pour produire notre modèle.

5 État de l'art : Classification de textures

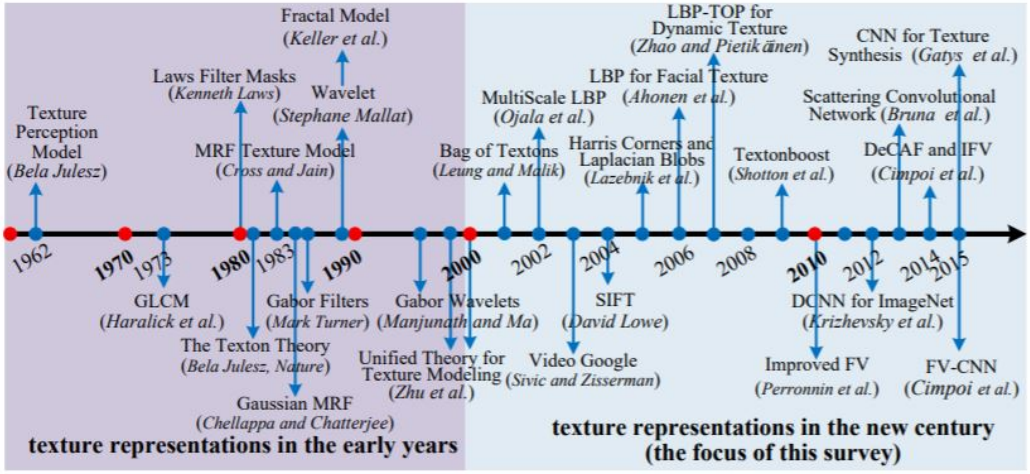


FIGURE 1 – Chronologie de la description de textures par Liu [8]

Avant l'utilisation de réseaux de neurones dans la classification de textures, d'autres approches étaient utilisées. Ces classifications étaient divisées en 4 approches différentes, les approches utilisées pour la description ou la classification de textures étaient divisés en quatre. Pour décrire les textures il existait quatre types d'approches différentes[9].

Une première approche consistait à décrire de manière statistique les textures, les histogrammes, les matrices de covariances mais également les motifs binaires locaux (qui est une approche aussi structurelle).

La seconde approche est structurelle et consiste à chercher des règles générant une texture, comme les micro-structures que sont les textons [9, chapitre 2].

Une autre approche est une technique utilisée en traitement du signal en passant dans le domaine fréquentiel avec des filtres de Gabor ou les wavelets.

La quatrième approche est une méthode basée sur des concepts tels que les modèles de fractales ou encore les modèles auto-regressif [9]. On retrouve ces types d'approches sur la première décennie de la chronologie élaborée par Liu *et al.* (voir Figure 1).

Plus tard, d'autres types d'approches basés sur des apprentissages pour décrire les textures sont arrivés comme les "sacs de mots", dont la représentation dépend de l'apprentissage d'un dictionnaire et dont la représentation dépend de l'occurrence du vocabulaire de ce dictionnaire dans l'image. Enfin, plus

récemment, des approches de classifications de textures à partir de réseaux de neurones convolutifs ont fait leur apparition.

6 Détection de défauts

6.1 Base de données

La base de données regroupe deux types de grilles, chaque grille possède 50 images sans défaut et 50 images avec défaut. La distance et l'éclairage des grilles ont été contrôlés lors de la photographie. Les données sont étiquetées sans défaut ou avec défaut. L'étiquetage du défaut n'est pas localisé. Les photos ont parfois été prises sur un fond noir et parfois sur un fond blanc.

Les grilles TM81_69 sont des grilles régulières où le défaut est difficilement visible à l'oeil. Lorsque ces grilles ont un défaut, elles ne sont pas parfaitement plates et sont bombées. Lors du contrôle qualité de celles-ci, les défauts sont décelés au toucher car il est difficile de les apercevoir à l'oeil nu, cependant lorsque les grilles sont bombées, une légère différence d'illumination est observée.

Les grilles TM48 ont un motif quadrillé régulier. Lorsqu'il y a un défaut, on observe une irrégularité dans le motif de la grille avec des lignes de quadrillage plus espacées que d'autres.

Notre approche consistera donc, pour un type de grille donné (parmi TM81_69 et TM48), à établir un modèle de classification (défaut *vs* sans défaut) basé sur l'apprentissage automatique. La Figure 4 illustre pour chaque type de grille un exemple d'image dans chacune des classes avec et sans défaut.

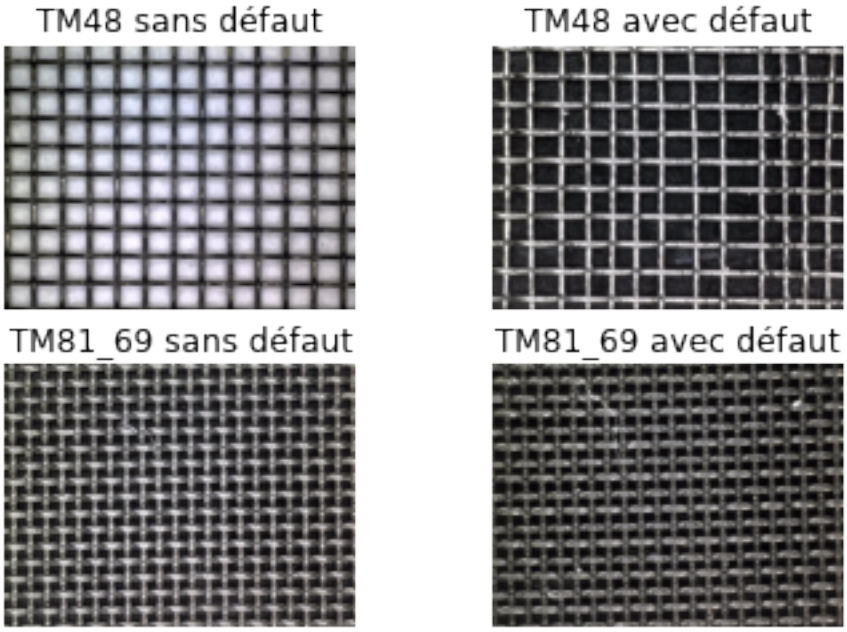


FIGURE 2 – Exemple des données grilles

6.2 Apprentissage

6.2.1 Introduction

L'apprentissage automatique consiste à définir les paramètres d'un modèle par la minimisation d'une fonction de coût représentant la somme des erreurs associées au modèle. L'apprentissage consiste à faire évoluer les paramètres de ce modèle pour atteindre un coût minimum. Dans ce but, un algorithme d'optimisation est utilisé pour se rapprocher de ce minimum en limitant les temps de calculs. En principe, si l'algorithme fonctionne normalement, au cours des itérations de l'algorithme d'optimisation, la somme des erreurs doit diminuer et la précision (généralement nommée *accuracy*) calculée sur les données d'apprentissage augmente en même temps. Dans le cas où le modèle n'est pas bien calibré, cette précision augmente sur les données d'apprentissage mais pas sur des données hors de cet ensemble (données de test). Il s'agit d'une forme d'apprentissage "par coeur" par l'algorithme que l'on nomme sur-apprentissage. Afin de visualiser les bon fonctionnement d'un algorithme on observe au cours des itérations ces valeurs (voir par exemple Figure 6).

L'apprentissage supervisé consiste à effectuer un apprentissage automatique avec une base de données étiquetées. Le modèle établit des frontières de décisions entre chaque étiquettes grâces aux différentes caractéristiques des données. Lorsque l'étape d'apprentissage est terminée, de nouvelles données peuvent être introduites dans ce modèle pour déterminer leurs étiquettes en fonctions de leurs caractéristiques et des frontières de décisions du modèle. Cette étape est nommée l'inférence. L'utilisation d'un système d'apprentissage est distinguée en deux étapes, la phase d'apprentissage et la phase d'inférence.

L'apprentissage par transfert consiste à utiliser un réseau déjà entraîné sur une autre base de données puis de l'affiner sur la base sur laquelle l'apprentissage doit être réalisé. Cela consiste à charger les poids au lieu de les initialiser aléatoirement. Dans certains cas les poids chargés pour l'affinage du réseau permettront de réduire le nombre d'étapes pour l'ajustement des poids et donc de réduire les temps d'entraînement.

En apprentissage profond, l'apprentissage repose sur l'activation ou non de neurones en fonction des entrées reçue et des paramètres appliqués, du poids appliqués sur ses entrées et du biais appliqué au neurone. L'activation d'un neurone est définie par une fonction d'activation généralement non linéaire. Á la sortie du réseau, le coût est calculé et les poids sont ensuite modifié via un mécanisme de rétro-propagation.

6.2.2 Architecture du réseau de neurones

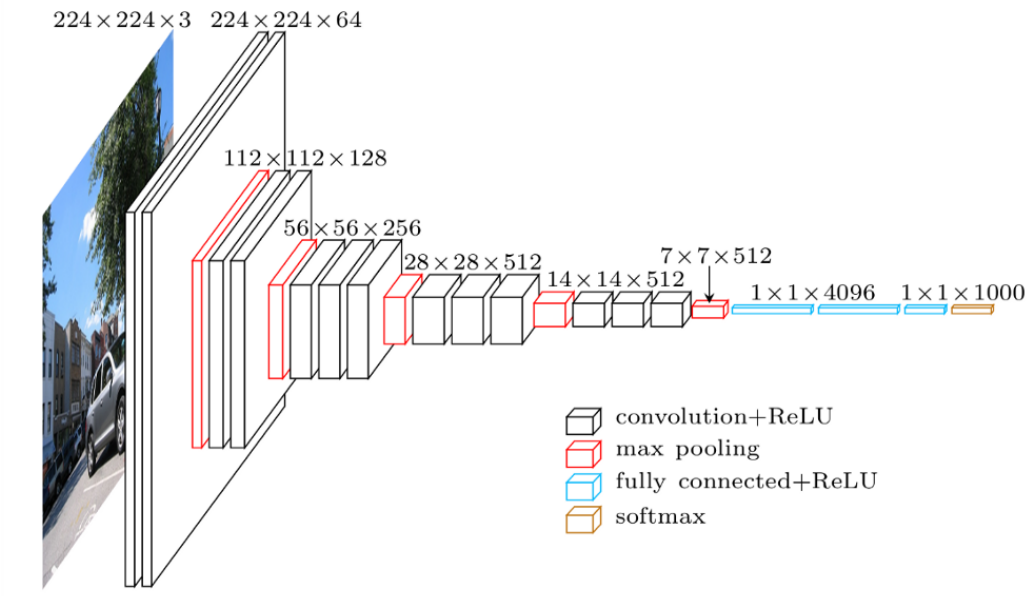


FIGURE 3 – Architecture du réseau VGG19 [1]

Un réseau de neurones convolutifs souvent cité en apprentissage supervisé a été choisi pour effectuer le classement qui a été réalisé avec le réseau VGG19 [11]. Il est composé de 19 couches de convolutions réparties en plusieurs blocs (Figure 3). Une couche de convolution est une couche consistant à réaliser une opération de convolution. Une convolution consiste à appliquer et à déplacer un filtre sur l'ensemble de l'image, chaque pixel du filtre étant un des poids du réseaux. Une couche de convolution fait généralement agir plusieurs filtres donnant pour résultats plusieurs cartes de caractéristiques.

Entre ces blocs de couches de convolutions, des couches de max pooling sont utilisées en terme de régularisation. Dans la fenêtre parcourant l'image lors de l'opération Max Pool, la valeur maximale dans cette fenêtre est conservée dans la nouvelle image qui sera réduite. L'objectif de la régularisation est d'éviter le surapprentissage du réseau c'est à dire d'éviter que le réseau se spécialise trop sur les données fournies en apprentissage ce qui empêcherait le réseau de généraliser le modèle. Suite à ces couches de convolutions, un réseau dense est activé par des fonctions ReLU. Celles-ci servent en partie à éviter le problème disparition des gradients[6] lorsque une fonction

d'activation saturante est utilisée. La saturation d'une fonction d'activation résulte en des gradients très petits qui peuvent disparaître dans les couches inférieures, ce qui nuit à la mise à jour des poids du modèle [6]. La fonction ReLU évite la saturation des valeurs positives mais lorsque sa sortie est négative la sortie de la fonction ReLU renvoie 0 et peut faire mourir le neurone [6].

L'apprentissage du réseaux a été réalisé par transfert en utilisant les poids du réseaux entraîné sur la base de données ImageNet[5]. La base de données ImageNet[5] est une base de données bâtie sur des images géométriques, c'est à dire que la classification se réalise surtout sur la forme d'un objet contrairement aux textures qui sont des motifs répétitifs pas forcément ordonnés.

Le réseaux a ensuite été affiné sur la base de données contenant les grilles.

L'affinage a été réalisé avec Adam avec un taux
d'apprentissage de 10^{-5} et des tailles de lots de 8.

6.2.3 Optimisation

Rétropropagation du réseau de neurones La mise à jour des poids d'un réseau de neurones est réalisée via un mécanisme de Rétropropagation. A la sortie du réseau l'erreur est calculée et l'erreur associée à chaque neurone est calculée en remontant les différentes couches du réseau, de la sortie vers l'entrée.

Descente de gradient La descente de gradient consiste à définir une suite de valeurs en calculant le gradient de la fonction de coût (dérivées partielles des différents paramètres et de suivre la direction donnée par le gradient de la fonction sur une certaine distance (appelée pas du gradient). La descente de gradient stochastique(SGD) est une descente de gradient où le gradient est calculé sur un exemple du jeu d'entraînement tiré au hasard. La descente de gradient par mini-lot est une descente de gradient où au lieu de calculer le gradient sur la totalité du jeux d'entraînement, il est calculé sur un lot du jeux de données.

L'optimisation nécessite un choix d' hyperparamètres (paramètres fixés "à la main" qui ne sont pas modifiés par le processus d'apprentissage automatique) pour permettre à l'algorithme de converger. Par exemple Un choix d'un taux d'apprentissage trop élevé empêche l'algorithme de converger et un choix de taux d'apprentissage trop bas ralentit la convergence de celui-ci. Afin de trouver le taux d'apprentissage on ajuste celui-ci, lorsque l'algorithme ne converge pas, le taux d'apprentissage est diminué.

D'autres algorithmes permettent de converger plus vite. La descente de gradient avec momentum permet d'accélérer la descente de gradient en allant de plus en plus vite vers le bas de la descente. Cette vitesse est mise en place par un vecteur qui est mis à jour après chaque itération. Un hyperparamètre permet de mettre une inertie pour rester en bas de la pente.

L'algorithme RMS Prop utilise taux d'apprentissage adaptatif c'est à dire que le taux d'apprentissage est adapté au fur et à mesure et devient de plus en plus petit pour se rapprocher de l'optimum. L'algorithme Adagrad réalise aussi un taux d'apprentissage adaptatif mais RMS Prop est plus efficace ne prenant que les derniers gradients en compte.

L'algorithme Adam (Adaptative Moment Estimation) mélange les idées de la descente de gradient avec momentum et de RMS Prop [6] et possède lui aussi un taux d'apprentissage adpatatif.

6.3 Traitement des données

L'utilisation d'une architecture pré-entraînée nécessite des images ayant les mêmes dimensions que la base de données avec laquelle elle a été pré-entraînée. Donc les images ont été redimensionnées avec une interpolation bilinéraire en 224x224.

Le classement a été réalisé dans un premier temps sans augmentation de données puis avec une augmentation de donnée transformant les images par des rotations, translation ou des symétries horizontales et verticales. L'augmentation de données a été réalisée en utilisant un générateur générant les rotations et translations entre 0 et une valeur maximale. Des rotations de 0 à 2 degrés ainsi que de 0 à 15 degrés ainsi que des translations jusque 10 % de l'image ont été testées. Les translations ont été effectués sur les deux axes horizontal et vertical.

Pour l'augmentation de données comportant des translations et rotations, les morceaux manquants de l'image résultants de la translations ou rotations sont remplacées par les morceaux coupés afin de répéter de manière périodique la texture de l'image.

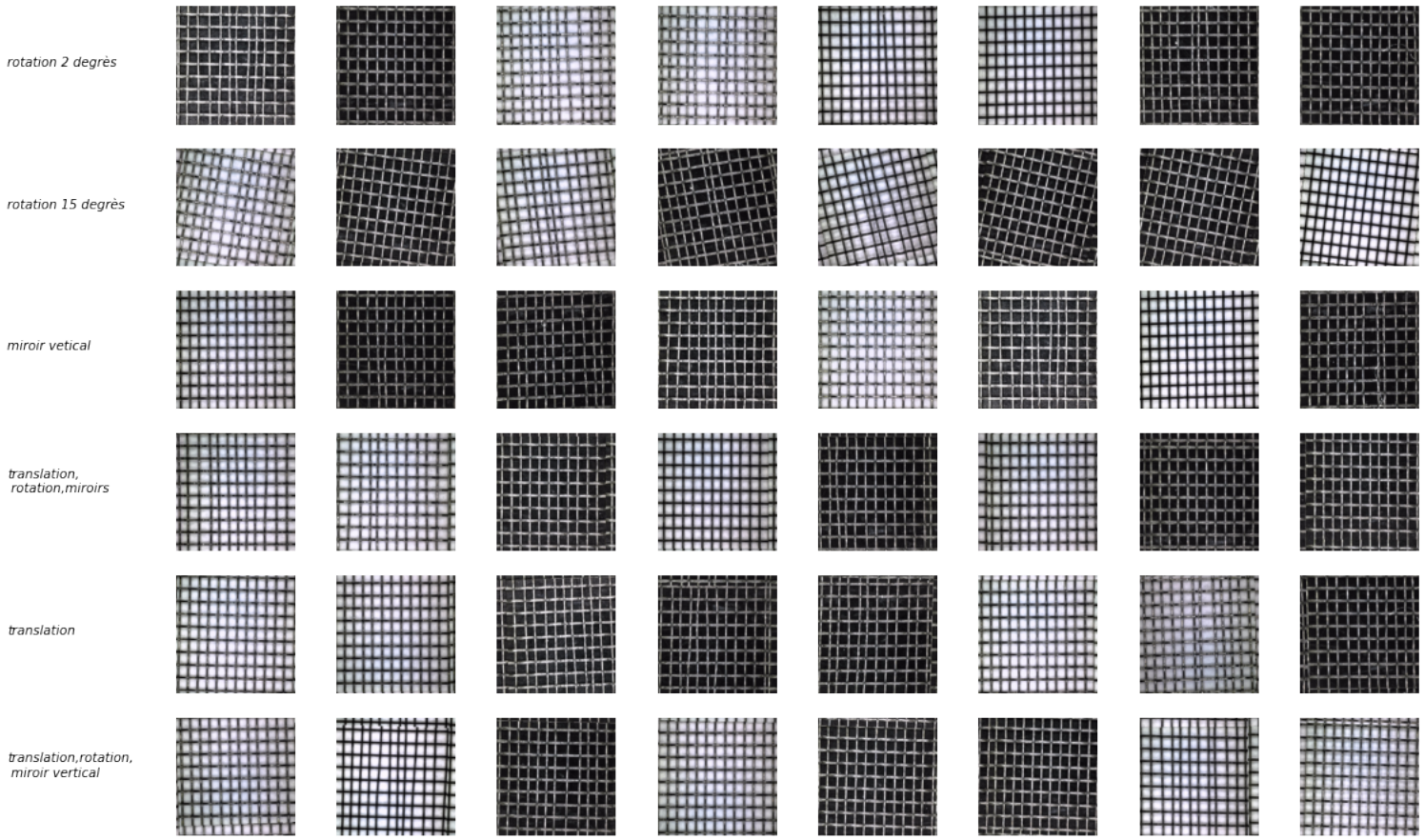


FIGURE 4 – Exemple d'augmentation de données générés pour un lot comportant 8 images

6.4 Validation

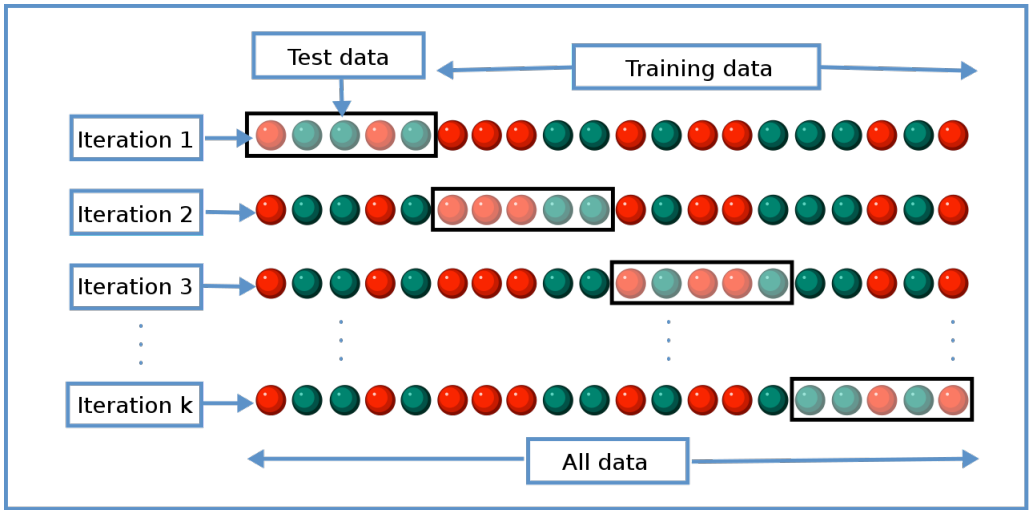


FIGURE 5 – K cross validation [2]

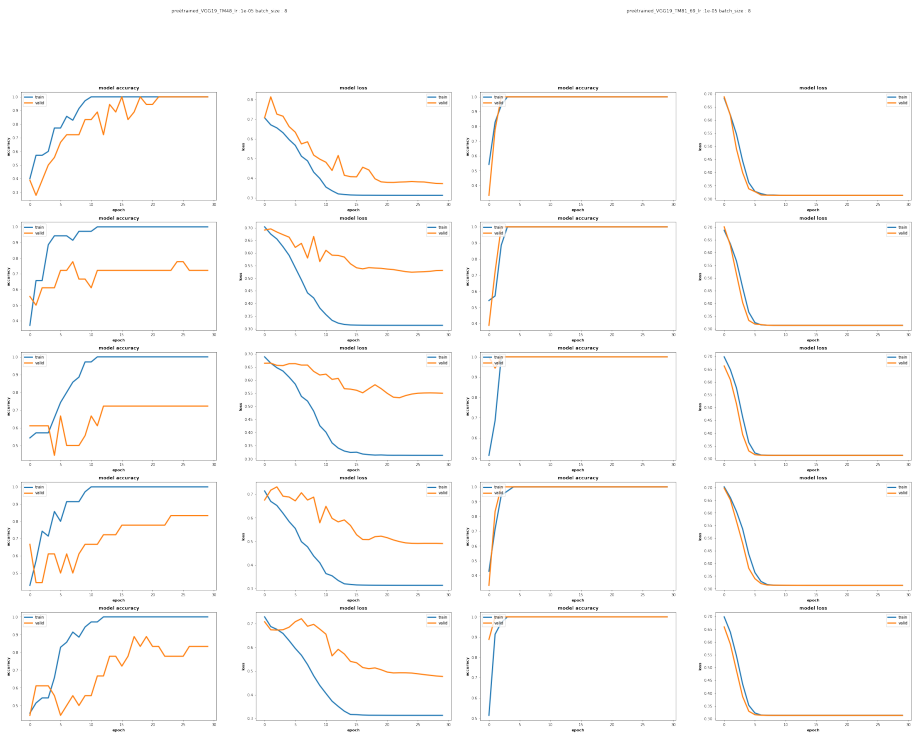
Afin d'éviter un surapprentissage on teste le modèle sur des données sur lesquelles il n'a pas été entraîné, ni servi à l'ajustement d'hyperparamètres. La base de données est petite impliquant donc de tester sur peu de données.

Pour cela la méthode de validation croisée a été utilisée avec 5 jeux d'entraînement/test. La méthode de validation croisée consiste à réaliser K combinaisons de jeux d'entraînement et test permettant de tester sur un plus grand ensemble d'images tout en conservant assez d'images pour l'entraînement. L'entraînement est recommencé 5 fois sur des données différentes et testé sur des données différentes. Les données ont été divisés en 5 jeux laissant 20 images test et 80 images d'entraînements à chaque ensemble. L'ensemble de validation représente 1/3 du jeu d'entraînement.

6.5 Résultats

6.5.1 sans augmentation de données

Avec VGG19 pré-entrainé sur imageNet on obtient en moyenne 15,8 images correctes sur 20 représentant une précision moyenne de 79% avec une variance de 3.36 (par rapport aux nombres d'images correctes) sur la grille TM48.



(a) TM48. (b) TM81_69

FIGURE 6 – Résultat d'affinage de VGG19 sur les grilles

6.5.2 Avec augmentation de données

Lorsque de l'augmentation de donnée a été utilisée, seules les données d'entraînement et de validation ont été augmentées. Les données test sont les images d'origines avec une dimension 224x224. A chaque lot, le modèle est testé avec 20 images, le tableau montre le nombre d'images correctes pour chaque lot. La précision moyenne est la moyenne sur l'ensemble de ces lots. L'augmentation de données a permis de stabiliser les différences de performance entre les différents lots, ainsi que la performance générale du modèle.

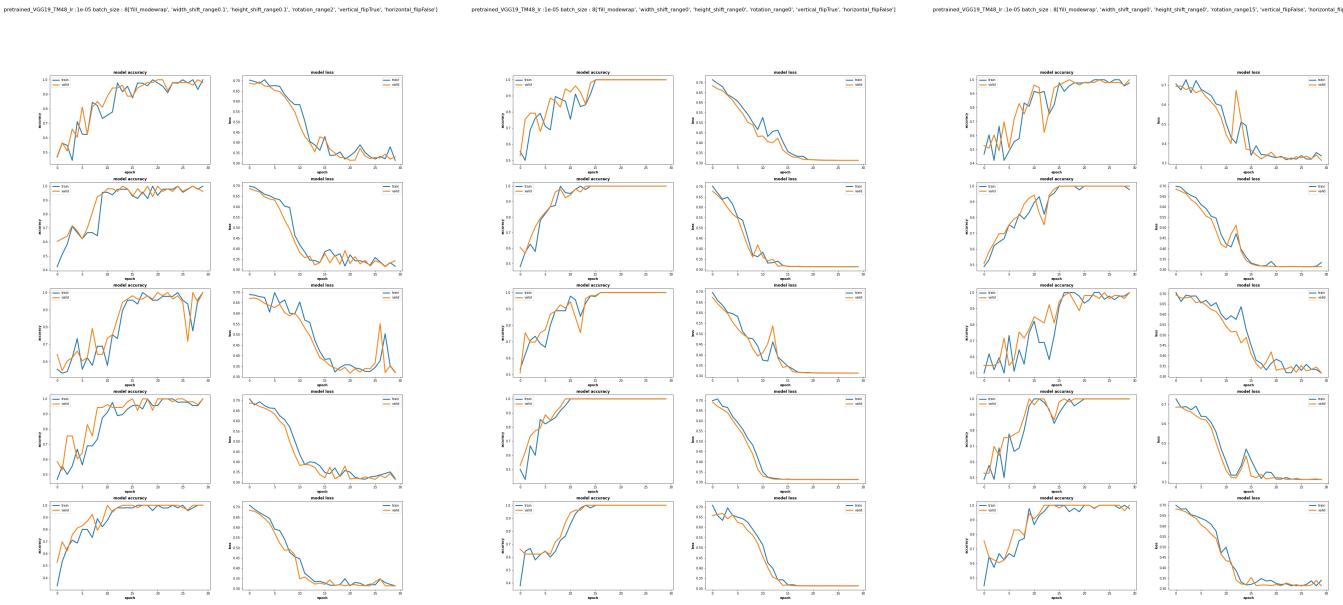
Augmentation de donnée	Précision moyenne (%)	s^2	Lot 1 (nb)	Lot 2	Lot 3	Lot 4	Lot 5
Sans	79 % (15.8)	3.36	14	18	14	15	18
Rotation 2 degrés	96%(19.2)	0.16	19	19	19	20	19
Rotation 15 degrés	96% (19.2)	0.96	20	18	20	20	18
Miroir vertical	96% (19.2)	0.96	18	20	18	20	20
Translation 10%	94% (18.8)	0.96	19	17	19	20	19
Translation 10%, Rotation 2 deg, Miroir vertical, Miroir Horizontal	96%(19.2)	0.16	19	20	19	19	19
Translation 10%, Rotation 2deg, Miroir Vertical	96%(19.2)	0.16	19	19	19	19	20

TABLE 1 – Résultats de l'entraînement sur la grille TM48

Augmentation de donnée	Précision moyenne (%)	Lot 1 (nb)	Lot 2	Lot 3	Lot 4	Lot 5
Sans	100 %	20	20	20	20	20
Rotation 2 degrés	100%	20	20	20	20	20
Rotation 15 degrés	100%	20	20	20	20	20
Miroir vertical	100%	20	20	20	20	20
Translation 10%	100%	20	20	20	20	20
Translation 10%, Rotation 2 deg, Miroir vertical,Miroir Horizontal	100%	20	20	20	20	20
Translation 10%, Rotation 2deg, Miroir Vertical	100%	20	20	20	20	20

TABLE 2 – Résultats de l'entraînement sur la grille TM81_69

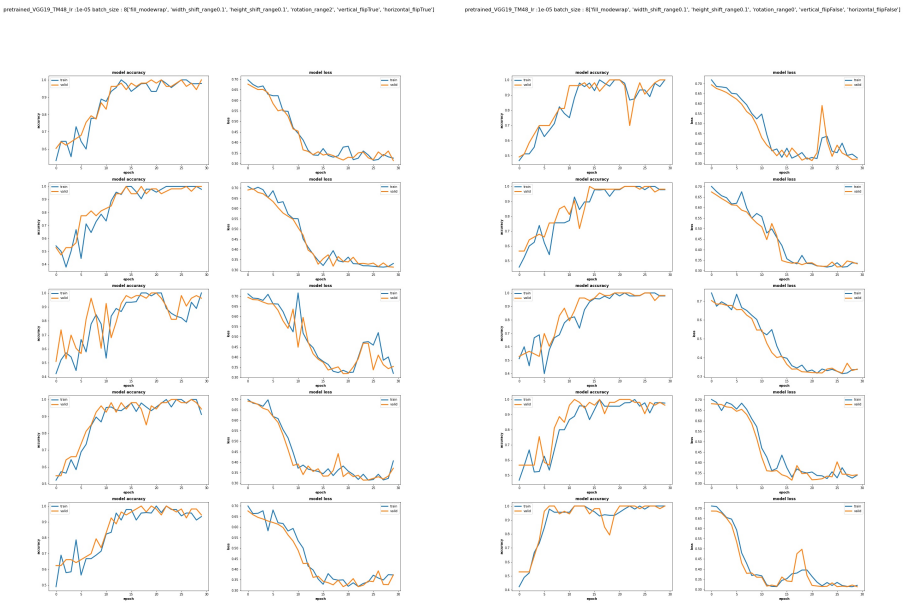
TM81 a des valeurs à 100% mais cela ne signifie pas que le modèle est parfait, pour chaque ensemble de test il y a seulement 20 images. Si la base de données avait été plus grande, la précision de 100% n'aurait pas pu être atteinte. Une erreur entraîne une chute de 5% et il n'est pas exclu que le modèle puisse faire des erreurs.



(a) miroir vertical,translation,rotation

(b) miroir vertical

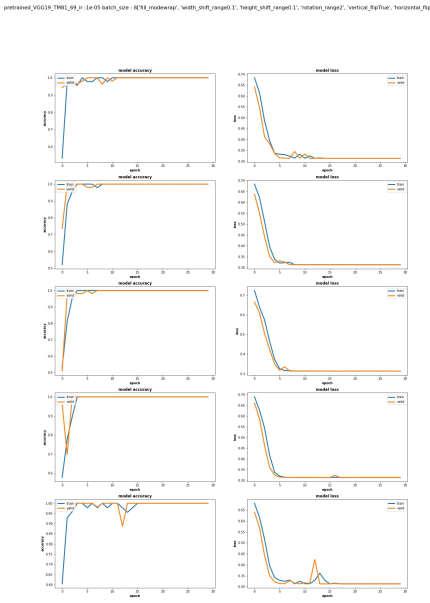
(c) rotation 15 degrés



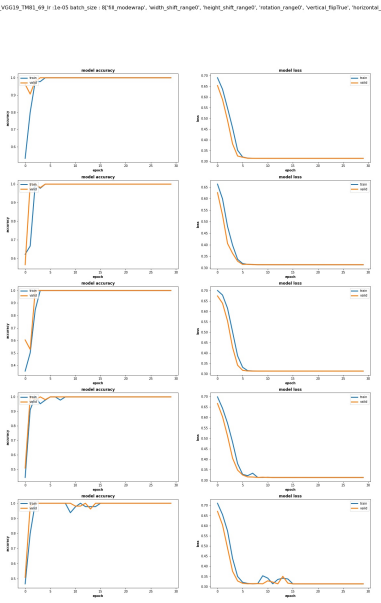
(d) Rotation, translation,miroir vertical et miroir horizontal

(e) Translation

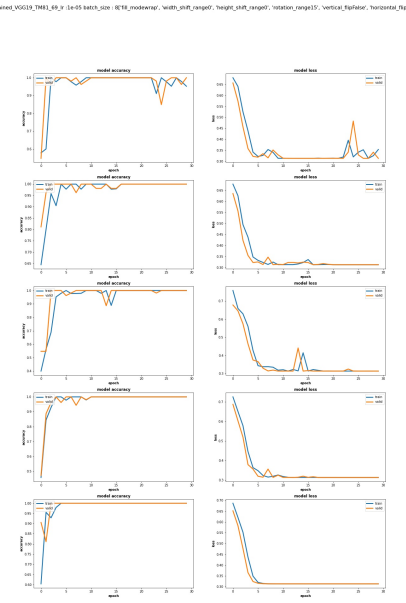
FIGURE 7 – Affinage de VGG19 sur TM48 avec différents paramètres d'augmentation de données



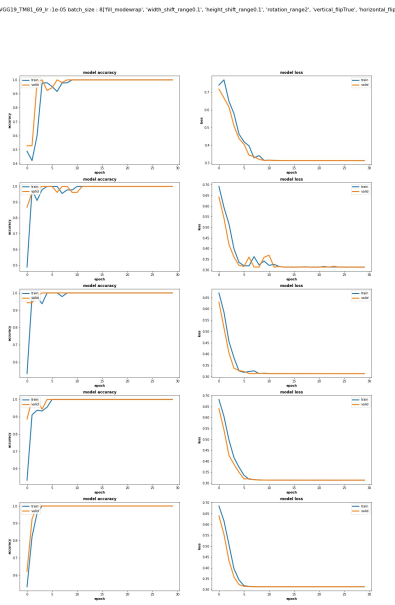
(a) miroir vertical,translation,rotation



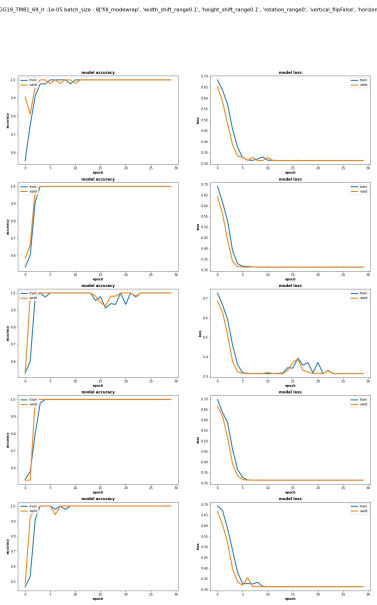
(b) miroir vertical



(c) rotation 15 degrés



(d) Rotation, translation,miroir vertical et miroir horizontal



(e) Translation

FIGURE 8 – Affinage de VGG19 sur TM81_69 avec différents paramètres d'augmentation de données

7 Textures à différentes échelles

Cette partie consiste à explorer des méthodes permettant de gérer une classification de textures à plusieurs échelles, et consiste à l'implantation du réseau Ganet [7]] et de sa pyramide d'échelles. Cette partie a été explorée en début de stage lorsque la base de données des tissages n'était pas encore réalisée.

7.1 GANet

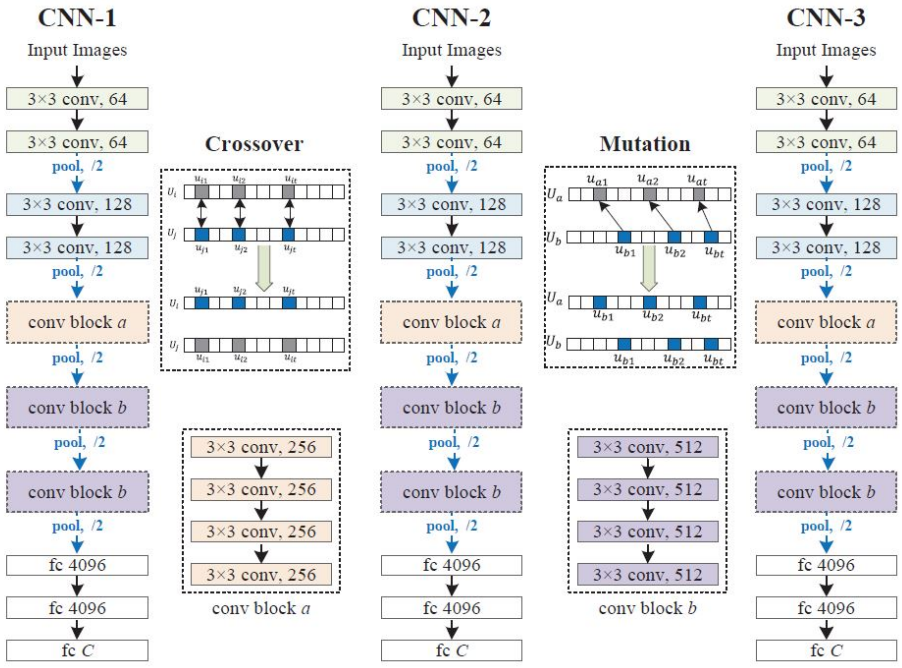


FIGURE 9 – Réseaux GANet [7] basés sur des algorithmes génétique entre 3 réseaux VGG19

Pour rendre l'apprentissage plus sémantique, GANet s'appuie sur l'entraînement de 3 réseaux VGGNet avec des opérations de cross-over et de mutation entre ces réseaux (inspirée des algorithmes génétiques). Afin de permettre ces opérations, les filtres de chaque couche de convolutions sont concaténées en une chaîne/séquence. Il est possible de se demander dans quel but un apprentissage sémantique est recherché dans le cadre de textures. Une supposition pourrait être la séparation de différentes textures ou

la séparation de la zone texturée de l'image. Comme le but de ce réseau est de pourvoir gérer différentes échelles. Dans la base de données, il y a des échelles où la texture en question ne représente pas l'intégralité de l'image.

Algorithme génétique Un algorithme génétique est un algorithme inspiré de la sélection naturelle. Lors de l'exécution de ce type d'algorithme, de nouveaux individus ayant leurs propres caractéristiques sont générés via des opérations avec la génération précédente.

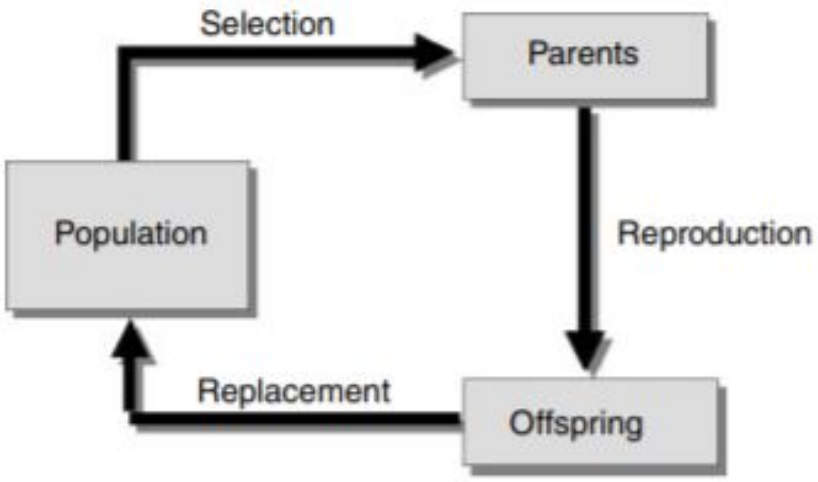


FIGURE 10 – Fonctionnement d'un algorithme génétique [3]

Le cross-over permet un mélange des caractéristiques entre deux parents à la génération suivante. La mutation est un changement spontané sur un individu d'une de ses caractéristiques. Chaque opération génétique a une probabilité de se déclencher. Une opération de sélection dont l'opération calcule le coût de chaque individu conserve les meilleurs individus pour la génération suivante, tous les individus ne sont pas conservés. [3] Les algorithmes génétiques présentés dans le réseaux GANet se trouvent être différents des algorithmes génétiques classiques. Au sein du réseau GANet il n'y a pas d'étape de sélection d'individu représenté par les différentes couches du réseau de neurones. Un chromosome dans le réseau GANet constitue une concaténation des filtres d'une couches de convolutions. Un modèle du réseau étant un individu possède donc 19 chromosomes représentant les 19 couches de convolutions. Le réseau GANet est considéré comme étant 3 in-

dividus représentés par les 3 modèles de réseaux de neurones convolutifs. Les opérations de mutations et de cross-over se réalisent entres les mêmes couches de convolutions entre ces réseaux.

7.1.1 Cross-over

L'opération cross-over se réalise en intervertissant des bits entre deux séquences de filtres piochées au hasard entre deux réseaux mais provenant de la même couche. Cette opération mélange les poids entre les couches de convolutions des trois réseaux. Ces trois réseaux ayant été entraînés sur des sous-ensembles différents de la même base de données, les poids de chaque réseaux sont différents.

7.1.2 Mutation

La mutation de cette architecture n'est pas une mutation comme généralement utilisée lors d'algorithme génétique. En algorithme génétique la mutation est une modification arrivant sur un chromosome/individu où ensuite un algorithme de sélection sélectionne qu'une partie des individus pour la génération suivante. Ici un chromosome est représenté par une concaténations des filtres d'une couche du réseau. Cependant la mutation diffère de celle des algorithmes génétiques classique dans le réseau GANet. Lors d'une mutation, un modèle (un réseau) copie certaines parties de filtres d'un autre modèle. Un seul des deux modèles est modifié. Pour l'opération de mutation, ces changements au niveau des filtres se réalisent de manière à ce qu'un filtre non-sémantique remplacent des parties de filtres sémantiques. Le but est de régulariser le réseau à la même manière d'une couche dropout. La couche dropout dans un réseau de neurones consiste à supprimer des neurones selon une certaine probabilité en prévention d'un surapprentissage. Dans un réseaux de neurones les couches les plus profondes sont plus spécifiques que les premières couches. La manière de déterminer si un filtre est sémantique dépend s'il s'agit d'une couche profonde ou non. Pour les premières couches si le motif est uniforme selon l'opérateur des motifs binaires locaux (LBP)[10] alors il est sémantique sinon il est non sémantique.

Le motif uniforme est un dérivé de l'opérateur binaire locaux et consiste à comparer le pixel central de la fenêtre avec le pixel à ses alentours. Si la soustraction entre le pixel gc au centre et le pixel gi (un de ses voisins) est positive alors le bit correspondant à ce voisin sera à 1. Une séquence binaire peut être définie pour l'ensemble des voisins de ce pixel situé au centre de la fenêtre par exemple 01011011. Le motif est uniforme si il y a moins 2 chan-

gements de 0 à 1 ou 1 à 0. Lorsque le motif est considéré comme uniforme alors cette région du filtre en question est considéré comme sémantique. Pour savoir si les filtres sont sémantiques ou non dans les couches plus profondes, les filtres sont visualisés et partitionnés en groupes sémantique et non sémantique. Cependant le façon décrite sur comment un filtre est considéré comme sémantique ou non sémantique n'est pas assez précise dans l'article GANet, ne permettant pas de savoir comment un filtre est considéré comme sémantique. Il est dit que l'activation minimum du groupe sémantique est considéré comme un taux permettant ensuite de déterminer des régions sémantiques dans les filtres cependant il n'y a pas d'informations sur comment trouver ce taux ou bien comment bien partitionner les filtres en deux groupes, sémantique ou non sémantique.

7.2 FV-CNN

D'après Cimpoi et al, les couches denses en sorties d'un réseau de neurones convolutif (CNN) capturent des informations spatiales pas nécessairement utiles aux textures ainsi que les couches profondes deviennent très spécifiques et deviennent moins transférable que d'autres couches [4]. L'architecture FV-CNN consiste à utiliser le réseau de neurones convolutif comme descripteur de l'image et ensuite utiliser une approche par "sac de mot visuel" pour enfin encoder par des vecteurs de Fisher.

Approche sac de mot L'approche "sac de mots visuel" consiste à créer un dictionnaire où le vocabulaire sont des caractéristiques d'images nommés patchs. La création de ce dictionnaire se réalise via un partitionnement des données (apprentissage non supervisé). Le vocabulaire ou les mots visuels de ce dictionnaire sont souvent des centres de classes du partitionnement réalisé. Le but est de pouvoir encoder et représenter les images en un vecteur basé sur ce vocabulaire ou en quantifiant l'occurrence de ce vocabulaire dans l'image. Il est possible par exemple de regarder l'occurrence du vocabulaire sur une image en réalisant un histogramme de mots visuels. Dans l'approche présentée avec les FV-CNN, le partitionnement des données est réalisé via un modèle de mélange de gaussiennes (Gaussian Mixture Model) et l'encodage est réalisé avec des vecteurs de Fisher. Le but de l'utilisation de l'approche par sac de mot visuel est d'avoir une représentation des données non ordonnées.

Le modèle de mélange de gaussiennes est un modèle génératif permettant de représenter les données par rapport aux densités de différentes gaussiennes.

L'encodage par les vecteurs de Fisher permet d'encoder dans une plus grande dimension 2KD, K étant le nombre de gaussiennes et D la dimension du descripteur de l'image.

8 Conclusion

Le but de ce projet était de réaliser une preuve de concept pour savoir s'il était possible de détecter des défauts sur les tissages avec la petite base de données fournie par Gantois. Avec un réseau classique il a été possible d'obtenir des résultats satisfaisant même si la petite taille de la base de donnée ne donne pas la plus représentative performance de celui-ci. Comme énoncé plus tôt, la performance du modèle sur les grilles TM81_69 ne peut pas être parfaite et une donnée seule peut tout de suite avoir plus d'impact sur la précision en positif ou en négatif. Cependant, les résultats montrent qu'il semble être possible de détecter des défauts, les modèles pourraient être améliorés sur de l'apprentissage sur des bases de données plus grandes.

Une évolution de ce projet pourrait être de réaliser un apprentissage qui localise les défauts grâce à une base de données contenant des boites englobantes (ou modèle de segmentation par exemple). De plus, d'autres approches pourraient être également explorées, comme par exemple la détection d'anomalies par l'apprentissage d'un modèle de tissages corrects (sans défaut). Une autre piste pourrait être la construction d'un modèle de tissage correct et de détecter les cas qui s'éloigneraient de ce modèle.

Références

[1] Image de l'architecture réseau de neurones vgg19 [11]. https ://mc.ai/extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models/.

[2] Image k-cross validation. https ://towardsdatascience.com/cross-validation-c4fae714f1c5 .

[3] Ajith Abraham. Evolutionary computation. In *Handbook of Measuring System Design*, chapter 131. American Cancer Society, 2005.

[4] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3828–3836, 2015.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet : A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] Aurélien Géron. *Deep Learning avec Tensorflow*. 2017.

[7] L. Liu, J. Chen, G. Zhao, P. Fieguth, X. Chen, and M. Pietikäinen. Texture classification in extreme scale variations using ganet. *IEEE Transactions on Image Processing*, 28(8) :3910–3922, 2019.

[8] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, and Matti Pietikäinen. From bow to cnn : Two decades of texture representation for texture classification. *International Journal of Computer Vision*, 11 2018.

[9] Majid Mirmedhdi. *Handbook of texture analysis*. Imperial College Press, 2009.

[10] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. volume 1842, pages 404–420, 06 2000.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 09 2014.

Table des figures

Liste des tableaux