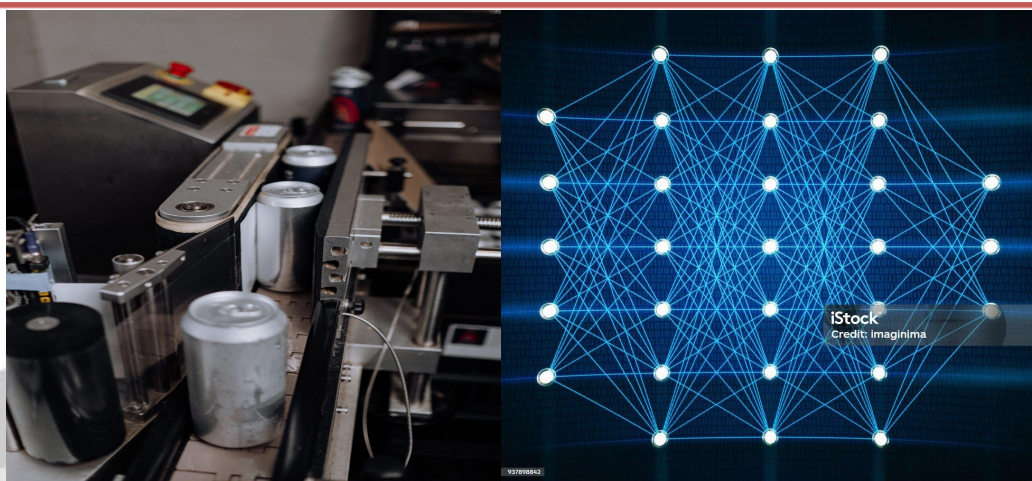




télécom
saint-étienne

école d'ingénieurs
nouvelles technologies

Comment reconnaître des défauts de fabrications grâce à un réseau de neurones convolutifs?



ANDRE Sanjiv
RETIF Louis
TIXIER Julien

Tuteur enseignant : MUSELET Damien

1. Introduction	4
2. Vision industrielle	4
2.1- Les sources lumineuses	4
2.2- Les techniques d'éclairage	5
2.3- La caméra et le traitement de l'image	6
3. Réseau de neurones	7
3.1- Conception d'un réseau de neurones	7
a) Le modèle du neurone	7
b) Réseau de neurones	8
3.2- Régression et classification	8
3.3- Fonction de perte (Loss function)	9
a) Mean Squared error	9
b) Cross entropy	9
c) Divergence de Kullback-Leiber	9
3.4- Fonction non linéaire	10
3.5- Fonction d'activation (activation function)	10
a) Fonction sigmoïde	10
b) Fonction tanh	10
c) Softmax	10
d) Fonction ReLu	10
3.6- L'entraînement du réseau	11
3.7- Les limites du réseau de neurone artificiel (ANN)	12
4. Le réseau de neurones convolutifs (CNN)	12
4.1- Architecture du CNN	12
4.2- Les couches de convolutions (convolutional layers)	13
a) Le kernel	13
b) Le padding	13
c) Le stride	14
4.3- La couche de correction ReLu	14

4.4- Les couches de pooling	15
4.5- Le flat layer	16
4.6- La couche fully-connected	16
4.7- Un exemple pour bien comprendre	16
<u>5. Conclusion</u>	<u>17</u>
<u>6. Bilan</u>	<u>18</u>
<u>7. Bibliographie</u>	<u>18</u>

1. Introduction

Avant d'aborder l'étude des réseaux de neurones artificiels et leurs applications à la détection des défauts industriels, il faut comprendre quels en sont les enjeux, les intérêts et comment l'intelligence artificielle a révolutionné la vision industrielle. Avant l'apparition et l'utilisation de cette dernière, le contrôle qualité était effectué par l'œil humain, c'était un travail fatigant, long et fastidieux voire même parfois impossible. Ainsi est apparue la vision industrielle qui n'est rien d'autre que l'automatisation des tâches de contrôle par une machine.

La vision industrielle c'est un système d'éclairage, une caméra pour remplacer les yeux et une unité de traitement pour remplacer le cerveau humain. Elle est utilisée pour vérifier la présence ou non d'un élément, sa position. Elle effectue également le contrôle de conformité, le contrôle d'aspect, s'il y a des rayures, des fissures à la fabrication de verres par exemple. Elle effectue également le contrôle dimensionnel, c'est-à-dire la vérification des dimensions comme la bonne longueur d'une vis, le comptage, l'identification, la surveillance... C'est au niveau de l'unité de traitement que le réseau de neurones sera implémenté par la suite afin d'améliorer grandement la détection des défauts en usine.

En effet, le réseau de neurones va apporter un réel changement. Premièrement au niveau de la flexibilité, les critères étaient avant fixés et les mesures faites à partir d'une référence. Le système va lui de son côté, pouvoir s'adapter aux variations dans les données sans l'intervention de l'homme. De plus, son analyse des défauts va être autonome, elle ne va pas nécessiter les critères définis par l'homme mais créer les siens, plus complets, se basant sur des éléments qui auraient pu nous échapper. Le réseau sera donc plus performant et va mener à la diminution des faux positifs et des faux négatifs.

2. Vision industrielle

L'éclairage est élément clé de la vision industrielle et ce en deux points importants: le type d'éclairage et la technique d'éclairage.

2.1- Les sources lumineuses

Le type d'éclairage peut se décomposer en trois catégories : les sources lumineuses incandescentes, fluorescentes et électroluminescentes.

L'incandescence vient de la propriété physique où tout solide ou liquide émet un rayonnement lumineux s'il est chauffé à plus de 700°C et plus la température augmente, plus l'intensité lumineuse sera forte. Ainsi, les lampes à halogènes sont composées d'un filament chauffé dans un milieu gazeux de la famille des halogènes. Ses lampes ont souvent une forte puissance qui est un avantage ainsi qu'une intensité lumineuse constante nécessaire lors de la capture vidéo pour un traitement de l'image uniforme et équivalent pour chaque produit de la chaîne. Néanmoins, les lampes halogènes n'émettent que de la lumière chaude ce qui peut parfois biaiser les résultats où être même carrément inutilisable dans certains cas si le produit observé absorbe les rayons. De plus, elles ont une faible durée de vie, environ 2000 h.

La fluorescence est une branche de la luminescence qui est l'émission de lumière à basse température sans incandescences. Certains corps peuvent émettre de la lumière lorsqu'ils reçoivent un rayonnement. Cette dernière est absorbée à une certaine longueur d'onde avant d'être réémise à une autre. Ainsi, un rayonnement ultraviolet est transformé en un rayonnement visible en excitant du phosphore dans les lampes fluorescentes. Leurs avantages sont qu'elles sont peu coûteuses, elles éclairent une grande surface et l'émission de lumière est uniforme. Cependant, elles ont une mauvaise stabilité à court et long terme. La luminosité varie selon la fréquence du réseau et son intensité diminue au fil du temps. Son éclairage est inhomogène et à une faible durée de vie, autour de 5000 h ce qui est tout de même deux fois plus que les lampes à halogènes.

Le dernier type d'éclairage est la source électroluminescente, c'est l'émission de lumière lorsqu'un semi-conducteur contenant un cristal et une jonction P-N est parcouru par un courant continu à basse tension. La jonction P-N est la zone où il y a la « frontière » entre une zone dopée positivement (P) et une zone dopée négativement (N) du même cristal. La source électroluminescente la plus courante étant la LED. Elle présente de nombreux avantages comparé aux autres sources, sa durée de vie est de 50 000h, son éclairage est stable dans le temps, elle peut être de différentes couleurs... En revanche la LED est assez coûteuse et peu puissante mais il y a une solution: le mode pulsé, on alimente la diode avec un très fort courant pendant un court instant ce qui engendre une très forte intensité lumineuse.

Les éclairages LEDs sont donc de plus en plus utilisés dû à leurs avantages et les inconvénients des autres types cités au-dessus.

2.2- Les techniques d'éclairages

De manière frontale, l'éclairage peut être directionnel, c'est-à-dire selon un angle. On placera la caméra en conséquence alors si on veut une réflexion diffuse ou spéculaire. En réflexion diffuse, le contrôle sera axé sur le contrôle des surfaces notamment brillantes. La réflexion spéculaire servira plus pour détecter les défauts de surfaces. Plus généralement, cette technique d'éclairage est utilisée pour le contrôle de présence et de mesure des pièces. Elle ne convient pas cependant pour les pièces transparentes et peut générer des ombres et des reflets.

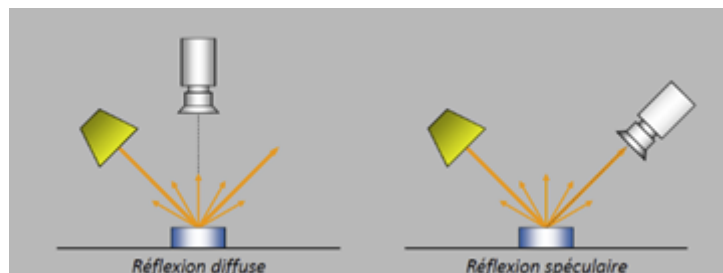


Figure 1 - éclairage épiscopique (Introduction à la vision industrielle)

Pour éviter cela on pourra alors utiliser l'éclairage coaxial qui présente les mêmes avantages sans ombres ni reflets, néanmoins elle demandera une plus grande intensité lumineuse.

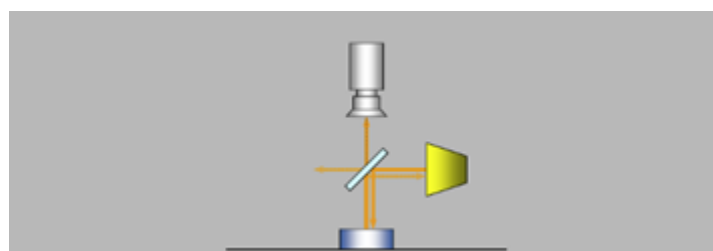
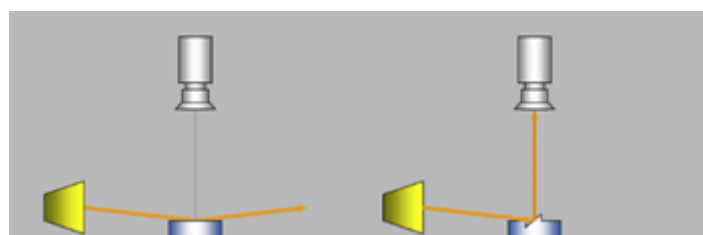


Figure 2 - éclairage coaxial (Introduction à la vision industrielle)

Pour un contrôle des défauts sur une surface, on emploiera un éclairage rasant. Il faudra également une forte source lumineuse.



ANDRE Sanjiv / RETIF Louis / TIXIER Julien
 2023

Figure 3 - éclairage rasant (Introduction à la vision industrielle)

A l'inverse, pour un contrôle de la forme d'une pièce et pour tout ce qui est translucide et transparent, on prendra un éclairage diascopique, la source lumineuse étant face à la caméra. On perd alors toutes les informations à la surface de la pièce.

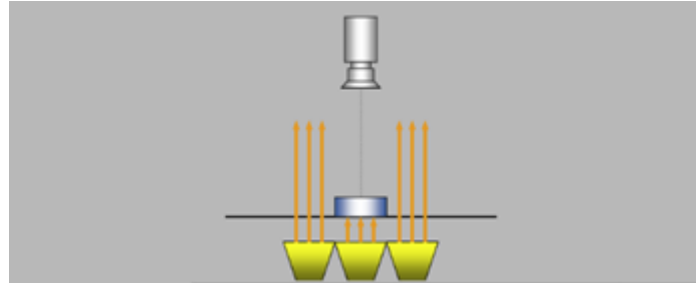


Figure 4 - éclairage diascopique (Introduction à la vision industrielle)

2.3- La caméra et le traitement de l'image

Une fois le choix de l'éclairage terminé vient le choix de la caméra, il faut qu'elle soit suffisamment performante pour avoir un traitement de l'image optimal par la suite. Néanmoins il faudra faire des compromis dû au prix de ses dernières. On parlera de caméra de manière générale mais on fera bien la distinction entre la caméra, l'appareil permettant de capturer une image et le capteur de vision industrielle qui ajoute à son bord sa propre unité de calcul. Souvent moins cher et plus facile à mettre en œuvre, il trouve ses limites dans ses applications dû à sa puissance de calcul limitée, dans notre cas l'implantation d'un réseau de neurones ne sera pas possible sur un capteur de vision.

Une des caméras couramment utilisée dans la vision industrielle est la caméra linéaire. C'est une caméra qui ne dispose d'un capteur que de 1xn pixels comme un scanner. Elle est très pratique sur une ligne de production dotée d'un tapis roulant en disposant d'une très grande fréquence d'acquisition.



Figure 5 - Illustration d'une caméra matricielle (à gauche) et d'une caméra linéaire (à droite) (Edmund Optics)

Il existe un bon nombre de caractéristiques pour choisir notre caméra. Dans le cas de la vision industrielle, les principaux sont :

- la résolution c'est-à-dire le nombre de pixels du capteur
- la fréquence d'acquisition autrement dit le nombre d'images acquises par la caméra en une seconde
- l'objectif

Le choix sera fait en fonction du cahier des charges. En effet, la fréquence d'acquisition nécessaire dépendra de la vitesse de la ligne de production et des besoins. L'objectif dépendra de la taille de la pièce à observer de la

distance à cette dernière qui dépendra des dispositions d'usine. La résolution dépendra du besoin en précision et également de la distance à l'objet observé.

Il faudra également faire attention aux différentes aberrations, l'aberration chromatique due à l'absorption des rayons lumineux suivant le matériau, la lumière... Mais également aux aberrations géométriques avec le phénomène de distorsion lorsqu'on s'éloigne du centre.

Avant de s'attaquer au réseau de neurones, parlons brièvement du traitement des images. Les intérêts sont d'atténuer le bruit, faire ressortir les contours, et compresser les informations sans en perdre afin de faciliter les calculs. Il existe trois types de modification d'une image:

- La transformation ponctuelle: le nouveau pixel dépend uniquement de l'ancien
- La transformation locale: le nouveau pixel dépend de l'ancien et de ceux qui l'entourent
- La transformation globale: le nouveau pixel dépend de l'ensemble des pixels de l'image.

3. Réseau de neurones

3.1- Conception d'un réseau de neurones

a) Le modèle du neurone

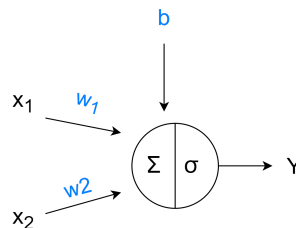


Figure 6 - Le neurone

En partant de l'idée que le cerveau humain peut réaliser des tâches complexes à partir de sa cellule maîtresse: le neurone, les chercheurs de l'époque se sont demandé si des architectures artificielles pouvaient être développées pour émuler cette capacité neuronale et accomplir des tâches similaires à celles du cerveau humain.

Architecture du neurone:

- Les entrées

Un neurone reçoit des signaux d'entrée x_i provenant d'autres neurones, de capteurs ou d'une couche précédente dans le réseau. Chaque entrée est associée à un poids w_i .

- Les poids

Chaque connexion entre une entrée et le neurone est pondérée par un poids w_i . Ces poids ajustent l'influence de chaque entrée sur la sortie du neurone. Pendant l'apprentissage, ces poids sont ajustés pour que le neurone puisse produire des sorties précises en réponse à des entrées spécifiques.

- Le biais

En plus des entrées pondérées, un neurone a souvent un biais b_i . Ce biais est un paramètre supplémentaire ajouté à la somme pondérée des entrées. Il permet d'introduire un ajustement dans la sortie du neurone.

- La fonction d'activation

Après avoir combiné linéairement les entrées pondérées et le biais, le neurone applique une fonction d'activation. Cette fonction non linéaire détermine si le neurone doit être activé ou non.

- La sortie

La sortie du neurone est la valeur résultante après avoir appliqué la fonction d'activation sur la somme pondérée des entrées et du biais. Cette sortie est transmise à d'autres neurones ou utilisée comme sortie finale du réseau pour des tâches telles que la classification, la prédiction, etc.

En combinant plusieurs de ces neurones dans des couches interconnectées, les réseaux de neurones peuvent apprendre des représentations hiérarchiques complexes des données en ajustant les poids et les biais. Cette architecture est à la base du fonctionnement des réseaux de neurones profonds.

b) Réseau de neurones

Maintenant que nous avons le modèle informatique du neurone biologique, nous pouvons pousser l'analogie un peu plus loin. En effet, le cerveau se constitue de centaines de milliers de neurones interconnectés. Ainsi, en s'inspirant à nouveau du mécanisme du cerveau humain, on connecte chacun des neurones artificiels pour former un cerveau artificiel, appelé réseau de neurones.

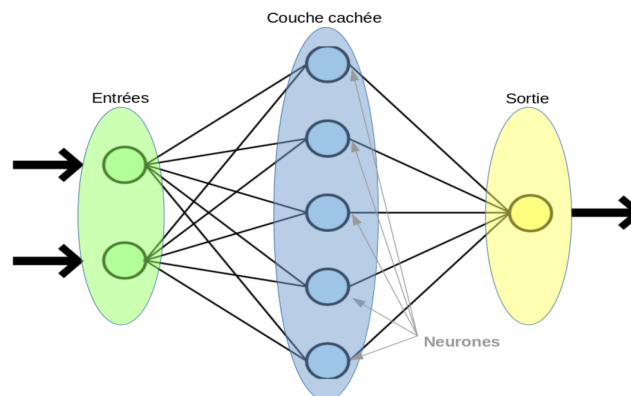


Figure 7 -Réseau de neurones

Malgré sa forte inspiration au cerveau humain, un réseau de neurones possède sa propre structure et sa propre manière de réagir face à un problème. Comme on peut le voir ci-dessus, un réseau de neurones se sépare en différentes couches.

La première est la couche d'entrée, sa fonction est de faire transiter les valeurs perçues par le réseau à tous les neurones de la couche suivante.

Les couches cachées ont pour fonction de calculer la somme pondérée de chaque entrée, d'y appliquer ensuite une fonction d'activation (sur la somme), pour enfin obtenir une nouvelle valeur qui sera aussitôt transmise à la prochaine couche.

Après avoir traversé l'ensemble des couches cachées du réseau, les données arrivent à la dernière couche appelée couche de sortie. Celle-ci aura les mêmes tâches à effectuer, mais elle appliquera en plus une fonction appelée "fonction de perte" afin de renvoyer la prédiction (le résultat) du réseau de neurones.

3.2- Régression et classification

Lorsqu'on cherche à résoudre un problème à l'aide d'un réseau de neurones, on a principalement deux approches. La première vise à réaliser des prédictions. Selon la valeur d'entrée, le réseau de neurones va prédire un résultat en cohérence avec sa base d'entraînement. En quelque sorte, il se demande quelle est la suite logique des résultats aux valeurs fournies lors de l'entraînement. Cette approche est obtenue par régression.

Lorsque l'on cherche à réaliser un classement de nos données par le réseau de neurones, la régression n'est pas adaptée, on opte plutôt pour une approche par classification. Cette dernière consiste à déterminer les

caractéristiques distinctives de nos valeurs d'entrée, et de renvoyer leurs appartenances (à quelle classe elles appartiennent) et leurs vraisemblances (probabilité que l'appartenance soit juste) en sortie. C'est cette dernière approche qui s'emploie pour les réseaux convolutifs et qui va être mise en évidence par la suite.

3.3- Fonction de perte (Loss function)

La loss function, ou la fonction de perte en Français, est l'outil qui nous permet de savoir si le réseau de neurones est performant ou non. Elle calcule la différence entre le résultat prédit et le résultat attendu pour une image donnée. Elle est donc très importante lors de la phase d'entraînement du réseau de neurone qui sera développé par la suite.

Il existe plusieurs fonctions de pertes qui ont chacune des caractéristiques qui leurs sont propres.

a) Mean Squared error

Dans un problème de régression, la fonction de perte choisie est généralement la fonction Mean Squared Error (MSN) en français erreur quadratique moyenne.

Expression:

$$MSN = \frac{1}{N} \sum_{i=1}^N (T_i - Y_i)^2$$

La MSN est une somme de carré qui peut donc s'interpréter comme la distance entre la valeur attendue et la valeur obtenue. Ainsi, plus le résultat est grand, plus notre erreur est grande. Par ailleurs, cette fonction est préférée à la simple valeur absolue de l'erreur, car elle évite des problèmes de sur-réactions.

Classification:

Quand on réalise un classement, plusieurs fonctions de pertes peuvent être utilisées. On utilise principalement la fonction de perte cross entropy ou la divergence de Kullback-Leibler.

b) Cross entropy

La cross entropy est une des fonctions de perte les plus utilisées dans les réseaux convolutifs, et ce pour deux bonnes raisons. En effet son expression est plutôt simple; elle applique l'opposé du logarithme sur des probabilités Y, les multiplie par T (les targets: probabilités visées), et réalise leurs sommes. Sachant que T vaut généralement 0 ou 1, on obtient des relations relativement simples. D'autre part, le gradient de la cross entropy est assez élevé (entre 0 et 1), accélérant ainsi l'apprentissage du réseau.

Expression:

$$CE = \sum_i -T_i \log(Y_i)$$

c) Divergence de Kullback-Leiber

La divergence de Kullback-Leibler est une fonction définie par deux distributions dont une qui représente les données (P), et l'autre qui sert de référence (Q). Sa fonction est de retourner la probabilité de vraisemblance d'une classe. Enfin cette fonction peut s'interpréter à tort comme la distance entre les valeurs des distributions.

Expression:

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

3.4- Fonction non-linéaire

Avant de définir les fonctions d'activation, il est important de préciser qu'elles doivent être non linéaires. L'intérêt est qu'elles n'entretiennent pas de rapports de proportionnalités, donc permettent une certaine sélection de leurs valeurs. C'est ainsi qu'on peut répondre à des problèmes de classifications.

3.5- Fonction d'activation (activation function)

Comme son nom l'indique, la fonction d'activation est la fonction qui va déterminer si son neurone associé doit s'activer ou non. Elle s'applique sur la somme pondérée des entrées et d'un biais, et renvoie la valeur en sortie du neurone. Son optimisation est donc primordiale dans le but d'avoir des résultats cohérents. Il existe plusieurs fonctions d'activations, ainsi selon le problème rencontré, on choisira la plus adaptée.

a) Fonction sigmoïde

La fonction sigmoïde prend son intérêt dans son domaine d'arrivée [0,1]. On peut ainsi interpréter ses résultats comme des probabilités.

Expression:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

b) Fonction tanh

Par rapport à la fonction logistique, Tanh converge plus rapidement et évite des problèmes de disparition de gradient. Elle demande ainsi une plus grande capacité de calcul.

Expression:

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

c) Fonction Softmax

La fonction Softmax renvoie une valeur comprise entre 0 et 1. D'autre part, si l'on prend la somme des Softmax en sortie d'un réseau de neurones, on obtient la valeur 1. Ainsi Softmax permet de retourner des probabilités, pour la classification par réseau de neurones. On verra plus tard que Softmax est la fonction d'activation utilisée à la dernière couche du réseau de neurones.

Expression:

$$\sigma(x)_j = \frac{e^{-x_j}}{\sum_{i=1}^N e^{x_i}}$$

d) Fonction ReLU

La fonction ReLU est une fonction très utilisée dans les réseaux de convolutions. Elle est linéaire sur l'intervalle $[0 ; +\infty[$ et constante à valeur 0 sur $]-\infty ; 0]$, autrement dit, elle donne x pour $x > 0$, sinon 0. De par ces caractéristiques, la fonction ReLU est utilisée comme un filtre. Elle est principalement utilisée dans les couches intermédiaires d'un réseau convolutif.

Expression:

$$ReLU(x) = \max(0, x)$$

3.6- L'entraînement du réseau

Pour apprendre, le réseau cherche à avoir les poids optimaux sur chacun de ses neurones pour avoir une frontière lors d'une classification la plus parfaite possible, le réseau va utiliser les loss functions, les fonctions qui mesure l'erreur avec un algorithme basé sur le gradient appelé souvent la descente de gradient. Dans le cas de la classification, la loss function est l'entropie croisée comme on l'a vu qui représente la fiabilité de notre réseau on cherche donc à la maximiser et pour cela on va utiliser la descente de gradient. Cette dernière se base sur la backpropagation ou rétropropagation en Français. On calcule les nouveaux poids en faisant les dérivées partielles par rapport aux fonctions précédentes et ainsi de suite en remontant le réseau, d'où le gradient.

Le gradient est très instable dû à la back propagation, étant donné que les couches basses sont les produits de toutes les couches supérieurs, il arrive que le gradient est de plus en plus petit en arrivant dans les couches basses du réseau ce qui va ralentir leurs apprentissages, c'est la disparition du gradient. Où à l'inverse dans l'explosion de gradient, ce dernier va augmenter et sera très grand dans les couches basses. Il est donc nécessaire de bien calibrer le gradient pour éviter ces problèmes mais également pour ne pas rester bloqué dans un minimum local, en effet la descente de gradient à pour objectif de minimiser l'erreur mais si on schématise notre erreur, elle ne représente pas seulement un puit avec un unique minimum mais plutôt une région vallonnée avec un minimum global et des minimums locaux, il est donc possible pour une descente de gradient de trouver un minimum local en pensant que c'est le global et cette erreur est dû au learning rate qui est trop petit. Ainsi son "pas" sera trop petit pour remonter de ce minimum local et chercher le minimum local. Mais il ne faut également pas avoir un learning rate trop grand au risque de ne pas trouver de minimum du tout. Pour cela il y a donc différentes méthodes pour effectuer une descente de gradient.

- Descente de gradient par lots

Dans la descente de gradient par lots, on va attendre une époque avant de mettre à jour notre réseau. c'est à dire que l'on va passer toutes les images de notre banque d'entraînement, additionner toutes les erreurs et à l'issue mettre à jour le modèle. Cette méthode apporte une stabilité au gradient et à la convergence mais elle risque justement de nous faire tomber dans un minimum local. De plus, plus la banque de données est grande, plus le temps de traitement est long car le système doit mettre en mémoire les données.

- Descente de gradient stochastique

A l'inverse, la descente de gradient stochastique met à jour les neurones après chaque image passée dans le modèle. C'est donc plus facile de mettre les données en mémoire et à beaucoup plus de chances de trouver le minimum global à la place d'un minimum local. Cependant elle va être moins efficace que la descente de gradient par lots et son gradient sera fortement bruité.

- Descente de gradient par mini-lots ou mini-batch

La descente de gradient par mini-batch est donc un entre-deux entre la descente de gradient par lots et la descente de gradient stochastique. Elle met à jour son modèle après chaque lot d'image passé

dans l'algorithme. Elle allie alors l'efficacité, la stabilité du gradient de la descente par lots et la rapidité de la descente stochastique. Elle en devient la méthode la plus utilisée pour ces raisons.

Il faut néanmoins faire attention au surentraînement ou overfitting en anglais car dans le cas d'une classification d'image, un réseau surentraîné ne va pas chercher à reconnaître l'objet mais à identifier cette dernière par rapport à sa banque de données ce qui n'est pas notre objectif. Les raisons d'un surentraînement sont nombreuses, cela peut provenir de l'architecture même de notre réseau, ce dernier étant trop complexe, il va se mettre à apprendre le bruit de nos images. Cela peut également arriver même si notre réseau est bien calibré si les images de notre banque de données sont trop bruitées, c'est-à-dire si elles possèdent trop d'informations inutiles. Le réseau sur-apprend s'il y a eu également trop d'époque, donc si la banque d'images est passée trop de fois dans l'algorithme. Il est aussi possible d'avoir un overfitting lorsque la banque de données est trop faible, on va alors perdre en précision et couvrir le moins de cas possible.

3.7- Les Limites du réseau de neurones artificiels (ANN)

Les ANN ont leurs limites lorsqu'il s'agit de travailler avec des images. En effet, dans un ANN, chaque pixel de l'image est directement lié à chaque neurone de la couche suivante, ce qui augmente massivement le nombre de paramètres à entraîner et rend le modèle très complexe et long en traitement. Par exemple, pour un ANN d'une couche de 1000 neurones, si on prend une image de 24 Mpx de dimension 6000*4000 px rvg, il y'aurait:

$$6000 * 4000 * 3 * 1000 = 72.10^9 \text{ paramètres}$$

C'est beaucoup trop. De plus, les ANN traitent chaque pixel indépendamment sans tenir compte de la proximité spatiale ou de la corrélation entre les pixels, ce qui est crucial pour comprendre les images.

Enfin, les ANN ne sont pas optimisés pour reconnaître des motifs locaux (bords, textures, formes...) qui sont essentiels dans la reconnaissance d'objets ou de caractéristiques dans une image.

C'est là que les réseaux de neurones convolutifs (CNN) entrent en jeu car ils sont spécialement conçus pour résoudre ces lacunes.

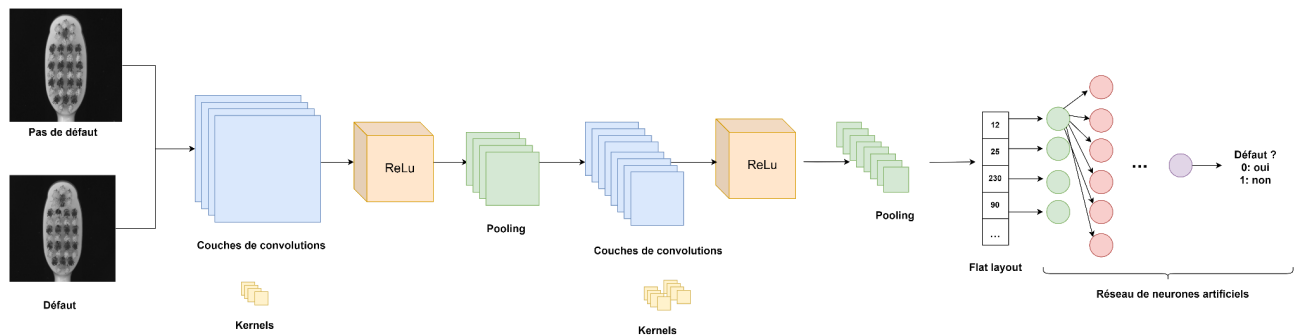
4. Réseau de neurones convolutifs (CNN)

4.1- Architecture du CNN

Le CNN est beaucoup plus efficace que les réseaux de neurones simples lorsque l'on s'intéresse aux images. Il est constitué de 4 couches:

1. Couche de convolution
2. Couche de correction Relu
3. Couche de pooling
4. Couche fully-connected (ANN)

A noter que les couches 1, 2 et 3 interviennent plusieurs fois à la suite. On peut également mentionner l'existence d'une couche 3bis: le flat layer (couche plane) qui permet de transformer des matrices $M_{n,p}(\mathbb{R})$ en une matrice $M_{n,1}(\mathbb{R})$.



LE CNN

Figure 8 - Le CNN

L'image en entrée contient beaucoup d'informations (plusieurs mégapixels), il va donc falloir extraire les informations les plus importantes de l'image grâce aux couches de convolution (formes, traits, textures...) pour obtenir des données qui auront une taille bien moins importantes et qui caractérisent au mieux l'image donnée en entrée. Ces données sont ensuite transmises dans le réseau de neurones dense qui va nous sortir un résultat, ici un booléen.

4.2- Les couches de convolutions (convolutional layers)

Une couche de convolution est l'une des couches fondamentales d'un réseau de neurones convolutif (CNN). Elle est responsable de l'extraction des caractéristiques de l'image. Dans notre CNN, plusieurs couches de convolutions seront utilisées pour faire ressortir différentes caractéristiques de l'image. Pour faire de la convolution, nous avons besoin d'un kernel (un filtre) et de notre image.

a) Le kernel

Un kernel est une matrice qui est utilisée pour effectuer des opérations de convolution sur une image. Le kernel agit comme une fenêtre glissante qui parcourt l'image et effectue des opérations locales sur les pixels. Les valeurs du kernel seront définies lors de la phase d'entraînement du CNN.

La fonction principale d'un kernel dans un CNN est d'extraire des caractéristiques spécifiques de l'image. Le kernel effectue une multiplication des valeurs du kernel par les valeurs correspondantes de la zone de l'image, suivie d'une somme de ces valeurs:

$$y = \sum_{i=1}^n \sum_{j=1}^m x_{i,j} \cdot \omega_{i,j}$$

Avec x une partie de l'image et ω le kernel.

La convolution avec un kernel permet de détecter des motifs ou des caractéristiques visuelles spécifiques dans l'image, tels que des bords, des textures, des formes, etc. Lors de la phase d'entraînement, le réseau va concevoir des kernels qui vont mettre en évidence un type spécifique de caractéristique.

Ainsi, en utilisant plusieurs kernels de différents types dans une couche de convolution, le réseau de neurones peut extraire une multitude de caractéristiques complexes et abstraites de l'image.

b) Le padding

Le padding est une technique utilisée pour traiter les bords des images lors de l'application du kernel. Il consiste à ajouter des 0 autour de l'image d'entrée avant la convolution.

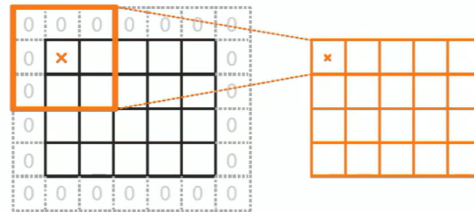


Figure 9 - Avec padding

Ainsi, ici après l'application du kernel 3x3 notre image garde la même dimension 5x4. Si il n'y a pas de padding, l'image est réduite et est de dimension 3x2:

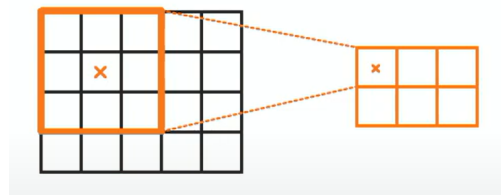


Figure 10 - Sans padding

Ainsi, le padding permet de contrôler la taille de la sortie de chaque couche de convolution, ce qui peut avoir un impact sur la manière dont l'information est préservée ou modifiée à travers les différentes couches du réseau. Il est souvent utilisé pour préserver les informations près des bords de l'image, améliorer la représentation spatiale des caractéristiques et pour s'assurer que la taille de l'image en entrée reste constante tout au long du processus de convolution dans un réseau de neurones.

c) Le stride

Le stride (pas) est un autre paramètre clé dans les opérations de convolution. Il contrôle le déplacement du kernel lors de la convolution à travers l'image d'entrée.

Le stride détermine le nombre de pixels de décalage à chaque déplacement du filtre lorsqu'il parcourt l'image. Plus précisément, il indique de combien de pixels le filtre se déplace horizontalement et verticalement à chaque étape de convolution.

4.3- La couche de correction ReLu

Tout neurone se doit d'avoir une fonction d'activation et le neurone convolutif n'y échappe pas, on utilise la fonction ReLu:

$$ReLU(x) = \max(0, x)$$

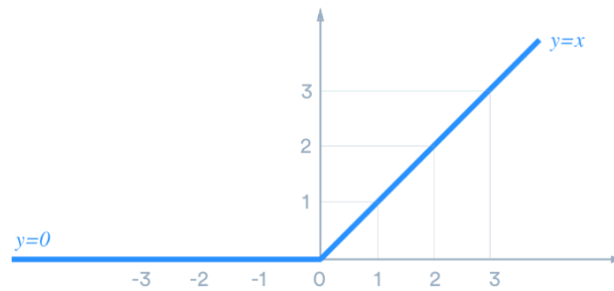


Figure 11 - Fonction ReLu

La couche de correction ReLu remplace toutes les valeurs négatives reçues en entrées par des 0.

La fonction ReLu va briser une partie de la linéarité en supprimant toutes les valeurs négatives, cela va donc accélérer les calculs.

Cette fonction permet au réseau de se concentrer sur les caractéristiques les plus importantes de l'image (valeurs positives) en laissant passer ce qui est utile et en éliminant ce qui ne l'est pas (valeurs négatives), contribuant ainsi à créer des représentations plus efficaces des données visuelles.

4.4- Les couches de pooling

Le pooling est utilisé pour réduire la taille de l'image. La couche de pooling permet de réduire le nombre de paramètres et de calculs dans l'ANN. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage. Il existe plusieurs types de pooling mais le plus utilisé est le max pooling. Le fonctionnement est assez simple: Pour chaque région (ici, une matrice 2x2), le max pooling sélectionne la valeur maximale parmi les valeurs de cette région et ne conserve que cette valeur maximale.

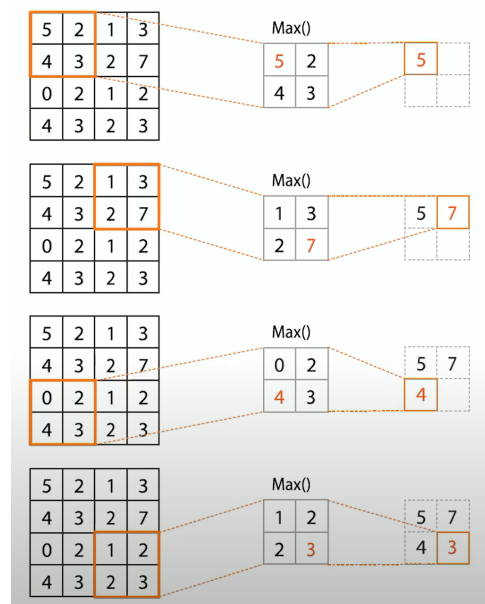


Figure 12 - Le pooling

Avec une fenêtre 2x2, on divise la taille de l'image d'origine par 2. A noter que la fenêtre se déplace avec un pas égal à sa largeur/longueur.

4.5- Le flat layer

On effectue ces 3 couches plusieurs fois (Convolution, correction ReLu, pooling), la taille de notre image va se réduire mais sa profondeur va augmenter (couches de convolutions). Les premières couches de convolution vont faire ressortir des bordures, des contours etc et les couches de convolution profondes vont récupérer des caractéristiques importantes de notre image, des grandes tendances etc.

A force de faire plusieurs pooling, notre image ne va faire que quelques pixels mais aura une profondeur importante. Avant d'arriver à l'ANN, on va donc "aplatir" l'image grâce au flat layer, on va transformer des matrices $M_{n,p}(\mathbb{R})$ en $M_{n,1}(\mathbb{R})$. On parcourt les matrices de gauche à droite.



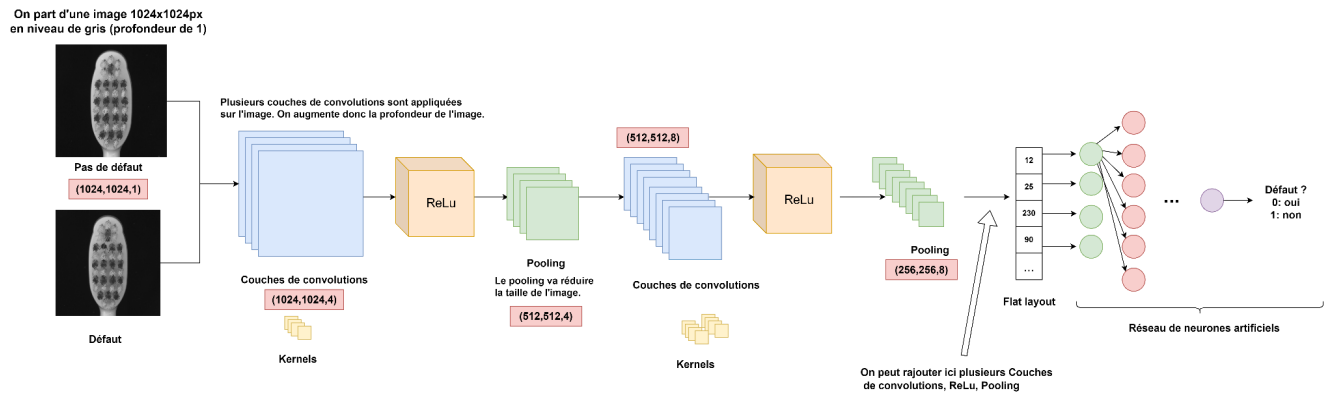
Figure 13 - Le flat layer

Cette couche est primordiale car elle permet de "mettre en forme" les valeurs avant de les entrer dans l'ANN.

4.6- La couche fully-connected

Les couches fully-connected sont les dernières d'un CNN, elles constituent un réseau de neurones denses sur lequel on va injecter les valeurs sorties par la flat layer. Ces valeurs arrivent alors dans les couches fully-connected (totalement connectées) du réseau convolutif, et comme expliqué précédemment (cf 3.1), sont soumises à une certaine fonction d'activation. En particulier, on applique à la dernière couche fully-connected (la dernière couche du réseau convolutif) la fonction d'activation softmax pour déterminer les probabilités des classes associées à l'image. Ainsi, l'ANN sera en mesure de nous prédire si oui ou non, il y a un défaut de fabrication.

4.7- Un exemple pour bien comprendre



LE CNN

Figure 14 - Le CNN

On part d'une image 1024x1024 en niveau de gris donc de profondeur 1, on a donc 1 048 576 pixels. Les couches de convolutions vont ajouter de la profondeur à notre image et le pooling va réduire la taille des images de convolutions. A la fin, dans l'exemple, on se retrouve avec une image 4 fois plus petite et avec une profondeur de 8. Autrement dit, on a 8 "petites" images de dimensions 256x256.

Ainsi, grâce au flat layout, on va mettre toutes ces images/matrices à plat, on obtient donc un tableau à une dimension avec $256 \times 256 \times 8 = 524\,288$ valeurs (pixels), soit 2 fois moins de valeurs qu'au départ et qui caractérisent au mieux l'image.

Ces données sont ensuite transmises au réseau de neurones dense qui va prédire si l'objet à un défaut de fabrication ou non.

5. Conclusion

En conclusion, le réseau de neurones convolutif apporte une vraie révolution pour les entreprises dans le domaine de la vision industrielle. Le réseau emmène avec lui une flexibilité au niveau du traitement de l'image ce qui permet de réduire légèrement les exigences entourant la vision, c'est-à-dire au niveau de l'éclairage et de la caméra. Il apporte également une autonomie en étant capable de s'adapter à des petits changements d'environnement et de conditions. Mais surtout, le réseau de neurones va améliorer les performances et la détection des défauts en créant ses propres critères qui sont parfois plus efficaces et plus poussées que ceux définissables par l'homme. Néanmoins pour obtenir la performance escomptée, le réseau nécessite d'un grand nombre de données et de beaucoup de temps pour se façonner. En effet, à la création du réseau, les poids sont fixés aléatoirement, il faut donc beaucoup de passage dans chaque neurones pour obtenir les poids optimaux. Mais il existe une solution, certaines entreprises ont développé des réseaux de neurones déjà entraînés sur une banque de données. Il suffit donc de l'adapter ou même parfois il est possible de l'utiliser directement.

Enfin les recherches et le travail effectué pour ce projet TIPE, nous ont permis d'explorer une petite partie du monde de l'intelligence artificielle, et ainsi, de forger une nouvelle critique vis-à-vis de ce dernier. En effet, avec la volonté de notre société de produire sans effort et de tout automatiser, on se rend bien compte que l'IA constitue un des plus grands enjeux de l'avenir. L'IA est capable de vaincre le meilleur joueur d'échecs, de copier le style des plus grands artistes peintres, ou de contrôler une voiture, mais ne se rapproche toujours pas du cerveau humain. En cause, les IA ont des applications très spécifiques et demandent une grande quantité de données, c'est le cas des réseaux convolutifs. D'autre part, une IA n'a pas de vie, ni même d'émotions, elle ne reste donc qu'une machine pour l'être humain.

6- Bilan

Objectifs Visés :

Le projet visait à explorer les capacités potentielles des réseaux de neurones convolutifs dans la détection des défauts de fabrication tout en se penchant sur les étapes intermédiaires (vision industrielle, réseau de neurone). L'objectif était de comprendre comment cette technologie pourrait être appliquée dans un contexte industriel pour améliorer la qualité et la précision de la détection des défauts. Nous avons atteint cet objectif.

Analyse du Projet :

L'étude approfondie du concept des réseaux de neurones convolutifs a permis de mettre en lumière leur potentiel dans la reconnaissance des défauts de fabrication. Cependant, étant une exploration théorique, le projet n'a pas impliqué la création ou l'implémentation réelle d'un réseau de neurones, nous restons donc sur notre faim.

Difficultés Rencontrées :

Le sujet, par sa nature même, s'avère être à la fois vaste et complexe. Aborder la détection des défauts de fabrication à travers les réseaux de neurones convolutifs est un défi nécessitant une compréhension approfondie des deux domaines, à la fois technique et industriel. De plus, la collecte d'informations pertinentes a posé un défi substantiel, en raison de la diversité et de la représentativité nécessaires pour valider théoriquement ces concepts.

7. Bibliographie

Defend intelligence: Coder un réseau de neurones convolutifs de classification d'image avec Python et Tensorflow. Disponible sur: <<https://www.youtube.com/watch?v=6FHtTyZxS5s>>

Wikipédia: Réseau neuronal convolutif. Disponible sur:
<https://fr.wikipedia.org/wiki/Réseau_neuronal_convolutif>

Chloé-Agathe Azencott - OpenClassroom. Utilisez des modèles supervisés non linéaires. Disponible sur:
<<https://openclassrooms.com/fr/courses/4470406-utilisez-des-modeles-supervises-non-lineaires>>

CNRS - Formation FIDLE. Réseaux convolutifs (CNN). Disponible sur:
<<https://www.youtube.com/watch?v=581X9wsnWJs>>

Morgan Gautherot. Tuto_MRI_ML. Disponible sur:
<https://github.com/MorganGautherot/Tuto_MRI_ML/blob/master/tp_1/Introduction%20au%20convolutional%20neural%20network.ipynb>

Lambert R. Focus: Le réseau de neurones convolutifs. Disponible sur:
<<https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/>>

Kimia Nadjahi et Pascal Monasse - OpenClassroom. Classez et segmentez des données visuelles. Disponible sur:
<<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles>>

ANDRE Sanjiv / RETIF Louis / TIXIER Julien
2023

MVTec - Download the MVTec AD dataset Disponible sur:

<<https://www.mvtec.com/company/research/datasets/mvtec-ad/downloads> >

J-M Dutertre - La jonction P-N Disponible sur:

<https://www.emse.fr/~dutertre/documents/1bis_diode_intro_semiconducteurs.pdf >

F.Gerossier - Introduction à la vision industrielle (GEII IUT UJM)

IBM - Qu'est ce que la descente de gradient?

<https://www.ibm.com/fr-fr/topics/gradient-descent>

Amazon - Qu'est-ce que le surajustement ?

<https://aws.amazon.com/fr/what-is/overfitting/>

Damien Muselet - Deep Learning for Computer Vision (Télécom St-etienne)

HEUDIN, Jean-Claude. Comprendre le Deep Learning. USA : Science eBook, 2017, 178.

Sources des illustrations:

Page de couverture: images libre de droits

Figure 7: datafuture: <https://datafuture.fr/post/fabrique-ton-premier-reseau-de-neurones/>

Figure 8 et 14: illustrations personnelles.

Figure 9-10-11-12 et 13: CNRS - Formation FIDLE. Réseaux convolutifs (CNN). Disponible sur:

<<https://www.youtube.com/watch?v=581X9wsnWJs>>



télécom
saint-étienne

école d'ingénieurs
nouvelles technologies

Résumé

Comment le réseau de neurones convolutif a apporté une révolution dans la détection des défauts de fabrication en étant plus autonome et plus performant? Comment ce dernier arrive à traiter l'information et comment les couches de convolution arrivent à extraire l'information pour arriver à ainsi détecter les défauts et les classer?

Afin de se rapprocher au mieux des capacités du cerveau biologique, les informaticiens chercheurs de l'époque eurent l'idée d'interconnecter les perceptrons entre eux. Ainsi le réseau de neurones est né. Son objectif est de répondre à des tâches plus ou moins complexes, comme par exemple superviser par imagerie une production industrielle.

Mots-Clés

Vision industrielle, deep learning, Intelligence artificielle, réseau de neurones, réseau de neurones convolutifs, industrie, informatique.

Abstract

How the convolutional neural network brings a revolution to manufacturing defect detection by being more autonomous and efficient? Exploring how it processes information and how the convolutional layers manage to extract relevant information, enabling the detection and classification of defects?

In order to get as close as possible to the capabilities of the biological brain, computer scientists and researchers of the time had the idea of interconnecting perceptrons between them. Thus the neural network was born. Its objective is to respond to more or less complex tasks, such as, for example, supervising industrial production using imaging.

Keywords

machine vision, deep learning, artificial intelligence, neural network, convolutional neural network, industry, computing.