

Article Incertitude des prédictions des réseaux de  
neurones dans l'Industrie Manufacturière: le cas du  
Grasping (Préhension Robotisée d'Objets)

jeffrey.verdiere

May 2021

## Résumé

La capacité d'automatiser la préhension d'objets ou de pièces est de plus en plus nécessaire sur les chaînes de production et dans un large nombre d'applications de l'industrie 4.0 qui requièrent de la robotisation. Un état de l'art sur l'implication de prédictions de zone de préhension sur des objets avec des réseaux de neurones (grasping par réseaux de neurones) a déjà été présenté dans la littérature. Néanmoins, peu de ces travaux s'intéressent réellement à la question de la sécurité et de la confiance que l'on peut avoir en ces structures neuronales en "vie réelle", c'est-à-dire sur les chaînes de production. Dans cet article, nous l'évaluation des incertitudes des prédictions des réseaux de neurones sera au centre de la problématique. La contribution de ce travail est de proposer une étape supplémentaire aux prédictions des réseaux de neurones présents dans la littérature traitant de la problématique du Grasping en y ajoutant un calcul de risque. L'objectif de ce calcul de risque sur les prédictions des structures neuronales présentées dans les recherches les plus récentes traitant de la problématique du grasping est de tester leur viabilité pour être implémentées en vie réelle. Cette étape supplémentaire mise en place et développée dans ce travail ne permet malheureusement pas d'alerter sur les mauvaises prédictions des réseaux impliqués dans les tâches de Grasping. Il faudrait tester d'autres méthodes exploratoires d'évaluation des incertitudes sur la problématique du Grasping.

# 1 Notations

$x_0, y_0$  : coordonnée en bas à gauche du rectangle de Grasping

$x_1, y_1$  : coordonnée du bas à droite du rectangle de Grasping

$x_2, y_2$  : coordonnée en haut à droite du rectangle de Grasping

$x_3, y_3$  : coordonnée en haut à gauche du rectangle de Grasping

$\sigma_{x0}$  : Incertitude sur  $x_0$

$\sigma_{y0}$  : Incertitude sur  $y_0$

$\sigma_{x1}$  : Incertitude sur  $x_1$

$\sigma_{y1}$  : Incertitude sur  $y_1$

$\sigma_{x2}$  : Incertitude sur  $x_2$

$\sigma_{y2}$  : Incertitude sur  $y_2$

$\sigma_{x3}$  : Incertitude sur  $y_3$

$\bar{\sigma}_{x2}$  : Incertitude moyenne sur  $x_2$  ou incertitude seuil

$\bar{\sigma}_{y2}$  : Incertitude moyenne sur  $y_2$  ou incertitude seuil

$\bar{\sigma}_{x3}$  : Incertitude moyenne sur  $x_3$  ou incertitude seuil

$\bar{\sigma}_{y3}$  : Incertitude moyenne sur  $y_3$  ou incertitude seuil

A : aire du rectangle prédite par le réseau de neurones

B : aire du rectangle labellisé

$x_{t0}$  : coordonnée de Grasping  $x_0$  : prédite par un réseau à l'issue du t ième masque de dropout

$x_{t1}$  : coordonnée de Grasping  $x_1$  : prédite par un réseau à l'issue du t ième masque de dropout

$x_{t2}$  : coordonnée de Grasping  $x_2$  : prédite par un réseau à l'issue du t ième masque de dropout

$y_{t0}$  : coordonnée de Grasping  $y_0$  : prédite par un réseau à l'issue du t ième masque de dropout

$y_{t1}$  : coordonnée de Grasping  $y_1$  : prédite par un réseau à l'issue du t ième masque de dropout

$y_{t2}$  : coordonnée de Grasping  $y_2$  : prédite par un réseau à l'issue du t ième masque de dropout

$y_{t3}$  : coordonnée de Grasping  $y_3$  : prédite par un réseau à l'issue du t ième masque de dropout

$$\textit{Indicateur de Jacquard} : IoU = \frac{A \cup B}{A \cap B} \quad (1)$$

## 2 Introduction et Contexte

D’après Costa et al. [2017] l’industrie 4.0 a un fort besoin d’automatisation de ses processus sur les chaînes de production pour répondre aux changements extrêmement rapides de la demande et aux besoins et à la concurrence.

Ainsi, d’après Bimont et al. [2020], les robots collaboratifs sont une des réponses à cette automatisation indispensable de l’industrie 4.0 et notamment de l’automatisation des chaînes de production. Ces robots doivent être, par exemple, capables de travailler avec des ouvriers, de les aider à soulever des charges lourdes ou à performer des tâches répétitives à faible valeur ajoutée. Des experts comme Kuka ont déjà proposé de nombreuses solutions à des problématiques très précises comme vérifier la taille d’une pièce ou translater des pièces de taille homogène sur un autre espace.

Néanmoins, Costa et al. [2017] considèrent que ces robots doivent pouvoir s’adapter très rapidement à travailler avec des objets de géométrie variable.

Ainsi, une des problématiques majeures selon Kleeberger et al. [2020] est la capacité des robots collaboratifs à pouvoir attraper des objets de géométries différentes. Cependant, contrairement aux humains qui lorsqu’ils sont confrontés à un nouvel objet sont capables de l’attraper immédiatement, les algorithmes actuels des robots collaboratifs ont de grandes difficultés à appréhender les géométries variables.

Schmidt et al. [2020] expliquent que les réseaux de neurones convolutifs pourraient remplacer les algorithmes de commande des robots collaboratifs pour répondre à cette problématique de grasping. Plus précisément, par leur capacité à s’améliorer et à s’adapter rapidement, les réseaux sont capables de s’adapter aux géométries les plus complexes et donc d’être très efficaces dans les tâches de grasping.

Dans ce contexte, Moon et al. [2020] affirment que les réseaux de neurones sont très performants dans de nombreuses tâches voire aussi performants que les hommes. [Krizhevsky et al.] expliquent que ces tâches peuvent aller de la classification d’images, de la reconnaissance d’objets, du traitement automatique du langage...

Moon et al. [2020] confirment que les réseaux de neurones sont très efficaces pour la reconnaissance de zones de préhension (grasping). Cependant, les mêmes auteurs affirment que la "surconfiance" de leurs prédictions posent de nombreuses limites dans leurs applications pratiques notamment dans les applications où la sécurité est essentielle.

L’objectif de cet article sera de combiner les méthodes les plus récentes de calcul d’incertitudes des prédictions des réseaux de neurones grasping impliquant des réseaux de neurones avec les méthodes les plus récentes de calcul des incertitudes des prédictions des réseaux de neurones. Plus précisément, il sera proposé une étape supplémentaire aux architectures neuronales sur des problématiques de Grasping impliquant un calcul de risque d’implémentation de ces structures "en vie réelle".

L’article présentera une revue de littérature sur les méthodes de grasping et sur le lien avec la question de la confiance que l’on peut avoir dans les prédictions des réseaux de neurones puis sera présentée la méthodologie de l’étude. Dans un deuxième temps, les résultats seront introduits. Enfin, ces résultats seront analysés au regard de la littérature scientifique et seront soulignées les limites de ces travaux de recherche avant de conclure. Plus concrètement, ce papier étudiera la méthode du Monte Carlo Dropout Weights (MC) pour estimer s’il est pertinent de mettre en place les méthodes actuelles de deep learning appliquées au Grasping en vie réelle.

### 3 Revue de la littérature

#### 3.1 Les différents types de Grasping

Saxena et al. [2006] expliquent que la préhension automatique d'objets à géométrie variable sur les chaînes de production est un enjeu majeur. En effet, les performances des robots sont encore très loin de celle des humains et cela reste un problème non résolu dans la plupart des champs de la robotique. Toujours les mêmes auteurs expliquent que beaucoup d'avancées ont eu lieu dans la définition des trajectoires mais peu de travaux ont réellement été concluants sur la détection de zones de Grasping. Ainsi, depuis les années 2010, un état de l'art conséquent propose différentes méthodes pour résoudre cette problématique de détection de zones de Grasping.

#### 3.2 Une définition de la detection de zone Grasping

Caldera et al. [2018] expliquent que le "Grasping" désigne la capacité d'un robot à détecter des zones où il peut l'attraper sans danger pour l'objet et pour son environnement.

Plus précisément, Kumra and Kanan [2017] expliquent que le problème du Grasping peut être formulé mathématiquement par la recherche d'un rectangle de grasping  $g$  sur l'image d'un objet. Ce rectangle de Grasping est une délimitation géométrique où un robot/pince a la possibilité de l'attraper.

$$g = f((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)) \quad (2)$$

où les  $(x_i, y_i)$  sont les sommets du rectangle de Grasping.

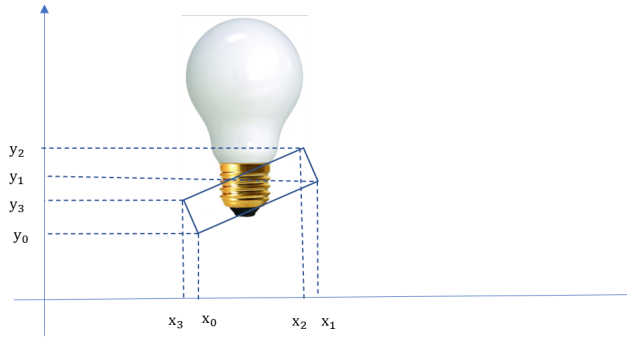


FIGURE 1 – Définition d'une zone de Grasping

L'objectif donc toujours selon Caldera et al. [2018] pour un algorithme de Grasping est de trouver le rectangle de Grasping le plus adapté à l'objet qu'un robot/pince doit attraper.

#### 3.3 Du problème de la détection de zones de Grasping à la nécessité du deep learning et des réseaux de neurones

Généralement, les robots effectuant des tâches de Grasping sont programmés manuellement par l'homme d'après Saxena et al. [2006]. Cela correspond donc à une série d'ordres effectués par un algorithme. Néanmoins, l'établissement de ces programmes est chronophage, notamment, lorsqu'il faut adapter les robots collaboratifs à attraper des objets de géométrie différentes ou complexes.

Kumra and Kanan [2017] expliquent que les résultats très probants des réseaux de neurones en traitement automatique du langage ou en vision par ordinateur ont poussé les chercheurs en robotiques à adapter ces structures de deep learning aux problématiques

de la robotique et notamment à la problématique de la detection de zones de Grasping. Toujours les mêmes auteurs expliquent que l'intérêt de l'application du deep learning dans le problème du Grasping est multiple :

- Gagner du temps sur la programmation classique des robots collaboratifs
- Traiter les géométries les plus complexes

### 3.4 De la nécessité de la détection de zones de Grasping par réseaux de neurones à la définition de la performance du Grasping

D'après Depierre et al. [2018], le critère utilisé pour savoir si un grasping est bien effectué ou non est le critère de Jacquard. Ce critère toujours d'après les mêmes auteurs consiste à vérifier que le ratio entre l'union et l'intersection entre le rectangle de grasping prédit et le vrai rectangle de Grasping est au dessus de 25%.

Cette métrique bien que très utilisée peut créer des "cas de faux positifs" où le critère de Jacquard est respecté mais d'un oeil humain la prédiction n'est pas forcément bonne. Néanmoins, toujours d'après les mêmes auteurs, cette métrique est largement utilisée dans la littérature du Grasping.

Ce critère de Jacquard est donné par la formule suivante :

$$IoU = \frac{A \cup B}{A \cap B} \quad (3)$$

avec A l'aire du rectangle de Grasping prédite par le réseau et B l'aire du rectangle de Grasping que le réseau aurait du prédire comme l'illustre la figure suivante :

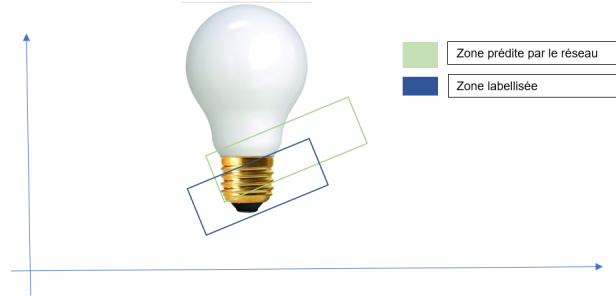


FIGURE 2 – Illustration du critère de Jacquard

D'une manière plus visuelle un exemple de bons résultats de Grasping est donnée sur la figure suivante :

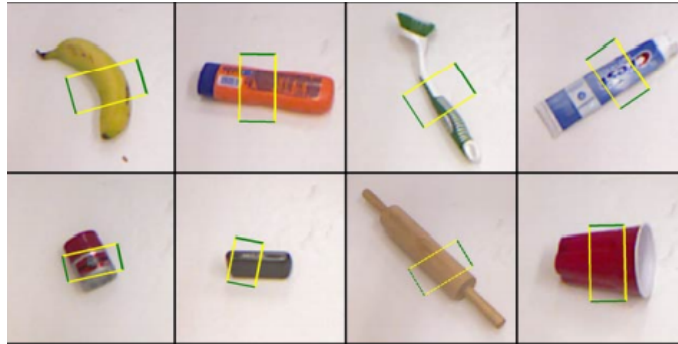


FIGURE 3 – Exemple de résultats corrects prédits par un réseau de neurones

### 3.5 Les performances des structures de réseaux de neurones de la littérature appliquées à la detection de zones de Grasping

#### 3.5.1 Apports d'une base de données pour entraîner des réseaux de neurones à effectuer des tâches de Grasping

D'après Redmon and Angelova [2015], la plupart des recherches étudiant les méthodes de Grasping faisant appel au deep learning, utilisent la base de données de l'université de Cornell. Cette base de données contient une multitude d'objets labellisés.

Pour être plus précis, Caldera et al. [2018] affirment que la base de données de Cornell contient 885 images de 240 types d'objets différents avec des labels de "bons" Grasping. Chaque image d'objet contient un rectangle de Grasping associé (modélisé par 4 points dans le repère cartésien dans un fichier texte). Ce rectangle de Grasping correspond donc à une zone où l'objet pourrait être attrapé.

Dans leur revue bibliographique des différentes méthodes de Grasping utilisant du deep learning, Caldera et al. [2018] affirment que le deep learning a permis d'obtenir de bons résultats sur la base de données de Cornell. L'état de l'art actuel montre l'amélioration des différentes architectures utilisées et a permis d'atteindre des précisions de l'ordre de 90%. La précision correspond au nombre de "bons grasping" effectués sur les images de validation. Un "bon grasping" est synonyme d'un critère de Jacquard respecté.

En effet, on peut observer les résultats des architectures les plus précises sur les 885 images de la base de données de Cornell dans le tableau suivant.

TABLE 1 – Evolution des performances des structures neuronales appliquées à la problématique du Grasping

Auteur	Structure de réseaux utilisée	Précision (Jacquard)
Jiang et al. [2015]	Fast Search	60.5%
Lenz et al. [2015]	Deep Learning	72%
Kumra and Kanan [2017]	SAE, Struct, reg.two-stage	73.9%
Asif et al. [2019]	STEM-CaRFs (Selective Grasp)	73.9%
Redmon and Angelova [2015]	Alex Net, MultiGrasp	88%
Asif et al. [2019]	STEM-CaRFs (Selective Grasp)	73.9%
Guo et al. [2018]	Hybrid-ach	89.1%
Chu et al. [2020]	VGG-16 and Res-50	90.5%

Même si les prédictions des réseaux sont meilleures d'année en année, comme illustré sur le tableau ci-dessous, il reste tout de même une erreur sur cinq de "mauvais" grasping et ce même avec les structures neuronales les plus récentes.

### 3.6 Des précisions limitées des architectures actuelles de réseaux de neurones appliquées au grasping à la question de la confiance

Comme l'expliquent Moon et al. [2020], malgré le pouvoir des réseaux de neurones dans un nombre vertigineux de tâches, notamment dans la detection de zones de Grasping, les réseaux ont tendance à être trop certains de leurs prédictions, ce qui pose un problème majeur lorsqu'il faut mettre en production ces structures de deep learning en "vie réelle".

D'autre part, Costa et al. [2017] expliquent qu'il est impossible de faire confiance à des réseaux de neurones qui agissent sur des chaînes de production qui n'ont pas un algorithme précis et fiable à 100%. Toujours, selon Costa et al. [2017], il n'est pas possible de faire confiance à un réseau de neurones qui est trop certain de ses prédictions alors que certaines d'entre elles peuvent être fausses ou que celui-ci n'atteint pas une précision de 100% pendant sa phase de test.



Néanmoins, les mêmes auteurs affirment que la seule solution actuelle pour adapter les chaînes de productions automatisées (robots collaboratifs) aux changements rapides de pièces ou de géométries, reste l'utilisation du deep learning. En effet, les réseaux de neurones sont essentiels pour pallier aux déficiences des algorithmes actuels.

### **3.7 Des précisions limitées des structures actuelles de réseaux de neurones appliquées au grasping à l'utilisation des méthodes d'évaluation des incertitudes**

Caldera et al. [2018] expliquent que des solutions pour améliorer les performances de ces réseaux de neurones seraient d'utiliser de plus grandes bases de données, d'utiliser des réseaux de neurones plus larges ou enfin d'utiliser d'autres structures de réseaux pré-entraînés. Néanmoins, toujours les mêmes auteurs affirment que ces possibilités permettraient d'augmenter au maximum de 10% la précision d'un réseau. Ce qui laisse une chance sur dix d'effectuer un mauvais Grasping.

Ainsi, Costa et al. [2017] expliquent qu'il est impossible de faire confiance à des réseaux de neurones qui agissent sur des chaînes de production qui n'ont pas un algorithme fiable à 100%. Néanmoins, les mêmes auteurs affirment que la seule solution actuelle pour adapter les chaînes de productions automatisées (robots collaboratifs) aux changements rapides de pièces ou de géométrie, reste l'utilisation du deep learning. En effet, les réseaux de neurones sont essentiels pour pallier aux déficiences des algorithmes actuels.

Ainsi, il pourrait être intéressant de mettre à profit les méthodes proposées par Gal [2016] pour évaluer les incertitudes des prédictions des réseaux de neurones dans le cas d'un apprentissage supervisé.

### **3.8 De la question de la confiance dans les architectures de réseaux de neurones au calcul de l'incertitude**

Deux types d'incertitudes/risques sont liées aux prédictions des réseaux de neurones.

#### **Epistémique**

Cette incertitude met en évidence l'ignorance d'un réseau de neurones (ce qu'il ne sait pas). En effet, les paramètres d'un réseau de neurones sont générés à partir de données d'entraînement qui ne couvrent généralement pas tout le problème que l'on souhaite résoudre.

#### **Aléatoire**

L'incertitude aléatoire met en évidence les bruits (erreurs de mesures) des données attribuées au modèle. Il peut s'agir des bruits issus de capteurs ou de mauvaises mesures

#### **3.8.1 Le calcul de l'incertitude épistémique**

Moon et al. [2020] expliquent que des approches bayésiennes donnent une représentation très simple de l'incertitude. Ainsi, avec une distribution sur les paramètres du réseau, différentes méthodes d'approximation bayésienne peuvent être utilisées pour approximer la distribution postérieure comme la méthode de Laplace [McKay, 1992], les chaînes de Markov [Neal, 1996] ou l'inférence variationnelle [Graves, 2011]. Ces méthodes sont très efficaces sur de petits réseaux mais pas sur de grands réseaux. Cela est très coûteux sur les réseaux modernes.

La méthode proposée par [Gal and Ghahramani, 2019] est plus adaptée aux structures de réseaux modernes (plus larges). Ils utilisent le Monte Carlo Dropout (MC) [Srivastava et al.] en phase de validation pour estimer l'incertitude en cachant aléatoirement les poids du réseau.

Ces mêmes auteurs expliquent qu’une fois l’entraînement du réseau réalisé avec des couches de dropout (utilisées originellement pour éviter l’overfitting), il est possible de réutiliser ces couches en phase de validation pour estimer l’incertitude.

Ainsi pour une même entrée  $x^*$  passée  $T$  fois dans notre réseau,  $T$  prédictions en sortie du réseaux sont obtenues. En effet, le dropout a pour effet d’annuler certains poids du réseau bayésien suivant une loi de Bernoulli de probabilité  $p_i$  et donc de générer un réseau de neurones différent à chaque tirage. Cette même entrée  $x^*$  passe donc par  $T$  réseaux bayésiens sensiblement différents, obtenus à partir d’un réseau initial. Cet effet de tirage est plus connu sous le nom de masque de drop out comme l’illustre la figure suivante.

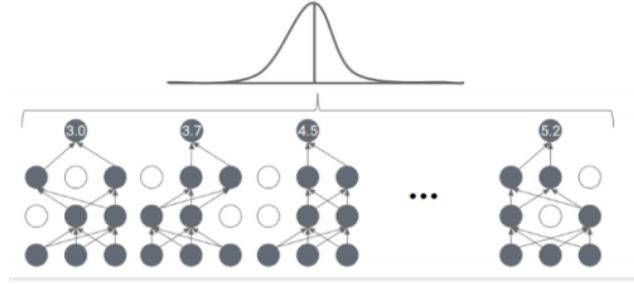


FIGURE 4 – Masque de Dropout

Ainsi, Weytjen and Weerdt [2021] montrent que le MonteCarlo Dropout est une méthode classique de régularisation stochastique qui évite aux réseaux de neurones de se surspécialiser durant les phases d’entraînement. Cela permet pour une même entrée d’ignorer les sorties de certains poids en multipliant chacun d’entre eux par un paramètre  $\epsilon$  échantillonné grâce à une distribution de Bernoulli de probabilité  $p$ .

### 3.8.2 Le calcul de l’erreur aléatoire

Kendall and Gal [2017] ont proposé une méthodologie permettant de décomposer l’incertitude en une incertitude aléatoire capturant le bruit des données et en une incertitude épistémique capturant les bruits du modèle. Même si ces méthodes réduisent fortement le coût de calcul pour estimer l’incertitude, cela est toujours très compliqué à mettre en place sur des réseaux plus larges.

Dans cet article, je propose de reprendre la structure neuronale Redmon and Angelova [2015] qui est très précise, de l'ordre de 90% et de mettre en confrontation cette structure aux méthodes les plus récentes d'évaluation des incertitudes des réseaux de neurones pour comprendre si il serait possible de mettre en place ou non cette structure en "vie réelle". L'objectif est multiple :

- Mettre en confrontation la structure neuronale de Redmon and Angelova [2015] qui est la plus récente et la plus précise avec les méthodes les plus récentes de Gal [2016] d'évaluation des incertitudes
- Vérifier s'il est raisonnable de mettre en place cette structure en vie réelle

C'est dans ce contexte, que mon travail explore les méthodes d'évaluation des incertitudes dans le cas précis des méthodes de grasping utilisées dans la robotique collaborative. Mon étude cherche à proposer une étape supplémentaire pour augmenter la sécurité des processus proposée par les auteurs actuels traitant de la problématique du grasping.

## 4 Méthodologie

### 4.1 Description du problème

Le problème consiste avant tout à estimer les incertitudes des prédictions des réseaux de neurones dans le cas précis des méthodes de Grasping impliquées dans la robotique collaborative. Plus précisément, pour une image d'objet, on souhaite obtenir le meilleur rectangle de Grasping possible. Pour cela, on entraîne un réseau de neurones à détecter des zones de Grasping (apprentissage supervisé) tout en calculant l'incertitude associée. L'objectif est donc d'obtenir un pourcentage d'incertitude sur les 4 coordonnées de Grasping que le réseau prédit.

### 4.2 La prédiction de la zone de grasping

#### 4.2.1 L'architecture

Pour construire une detection efficace de zone de Grasping, Redmon and Angelova [2015] utilisent cinq couches de convolutions suivies de 3 couches complètement connectées. Les couches de convolutions sont séparées par des couches de normalisation et de max-pooling. Une description plus précise est donnée sur la figure 5 suivante.

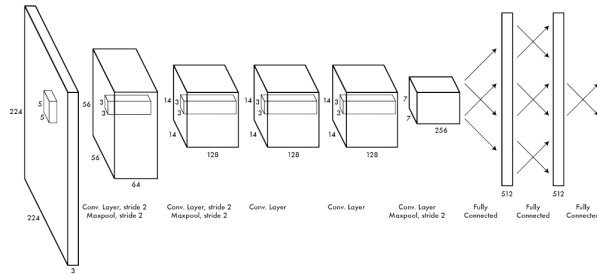


FIGURE 5 – Structure neuronale utilisée

#### 4.2.2 Une regression direction pour effectuer des tâches de Grasping

L'architecture présentée ci-dessus est une regression directe. Les images de la base de données de Cornell sont données au modèle qui utilise ces couches de convolution pour extraire des caractéristiques de l'image. Les couches fully connected se terminent par un output de sortie constitué de 4 neurones de sortie correspondant aux quatre coordonnées de grasping.

Nous faisons donc l’hypothèse que chaque image contient un objet ”attrapable” et qu’il existe une seule façon d’attraper l’objet. Il s’agit d’une hypothèse qui potentiellement ne serait pas valable sur une chaîne de production.

### 4.3 Mise en place de la base de données

Il a fallu d’abord récupérer les fichiers dans la base de données de l’université de Cornell. Les fichiers sont regroupés comme expliqué sur la figure 6 en une image avec son rectangle de Grasping associé (Quatre coordonnées dans le plan cartésien) sous forme d’un fichier texte comme le résume la figure 6 ci-dessous :

```
Image files
Named pcdxxxxr.png
where xxxx ranges from 0000-1034
These are the original images of the objects

Handlabeled grasping rectangles
Named pcdxxxxcpos.txt for positive rectangles
```

FIGURE 6 – Explication de la constitution de la base de données de Cornell

La base de données initiales de Cornell contient 884 images. Pour augmenter le nombre de données, [Shorten and Khoshgoftaar, 2015] proposent une symétrie axiale. Dans cet article, est utilisée une symétrie axiale par rapport à l’axe y et x pour chaque image et son rectangle de grasping associé ce qui permet de tripler la base de données pour arriver à 2652 images et rectangles de Grasping associés. On utilise alors le code couleur suivant : vert pour les rectangles en sortie du réseau et bleu pour les rectangles que l’on souhaiterait obtenir. On garde 2552 images et rectangles pour l’entraînement et 100 images et rectangles associées pour la validation.

TABLE 2 – Base de données de l’université de Cornell récupérée

Répartition données	Symétries effectuées	Après data augmentation
10 images de dentifrice	Symétrie par rapport à l’axe x Symétrie par rapport à l’axe y	30 images de dentifrice
10 images d’appareils photo		30 images d’appareils photo
10 images de cure dent		30 images de cure dent
10 images de grandes sucettes		30 images de grandes sucettes
10 images de téléphones		30 images de téléphone
10 images de craies		30 images de craies
10 images de spatules		30 images de spatules
10 images de canettes		30 images de canette
10 images d’appareils photo		30 images d’appareil photo
10 images de lunettes de soleil		30 images de lunettes de soleil
10 images de livres		30 images de livres
10 images de barres de chocolat		30 images de barre de chocolats
10 images d’ampoule		30 images d’ampoule
10 images de bols		30 images de bols
...		...
884 images		2652 images

On répartit ensuite la base de données en 2552 images d’entraînements et 100 images de validations comme illustré dans le tableau ci-joint :

Nous effectuons un minimum de preprocessing des données avant d’entraîner le réseau. Les images sont centrées et les pixels normalisés entre 0 et 1.

TABLE 3 – Répartition des données d’entrainement et de validations après Data Augmentation

Répartition données d’entrainements	Reparation données de validations
30 images de dentifrice	5 images de dentifrice
30 images d’appareils photo	5 images d’appareils photo
30 images de cure dent	5 images de cure dent
30 images de grandes sucettes	5 images de grandes sucettes
30 images de téléphones	5 images de téléphone
30 images de crais	5 images de crais
30 images de spatules	8 images de spatules
30 images de canettes	7 images de canette
30 images d’appareils photo	5 images d’appareil photo
30 images de lunettes de soleil	5 images de lunettes de soleil
10 images de livres	5 images de livres
10 images de barres de chocolats	5 images de barre de chocolats
10 images d’ampoule	20 images d’ampoule
10 images de bols	10 images de bols
...	10 images de bols
2552 images d’entrainement	100 images de validation

#### 4.4 L’entrainement du réseau de neurones

Comme dans l’article de Redmon and Angelova [2015] le modèle est entraîné sur 25 apprentissages sur l’ensemble des données et avec un taux d’apprentissage de 0.0005 sur l’ensemble des couches ainsi qu’un weight decay de 0.001. On met également en place des couches de dropout entre les couches de convolutions de probabilité  $p=0.5$ .

TABLE 4 – Paramètre d’entrainements du réseau

Taux d’apprentissage	Weight decay	Dropout	Nombre d’epochs
0.0005	0.001	0.5	35

#### 4.5 Le calcul de l’incertitude

Pour récupérer l’erreur épistémique dans un réseau de neurones (NN), on place une distribution de probabilité sur les poids du réseau. On place une gaussienne :  $W \sim N(0, I)$ . Cela permet d’après Weytjen and Weerd [2021] d’ignorer de manière aléatoire certaines sorties du réseau en multipliant chacun des poids du réseau par un paramètre  $\epsilon$  échantillonné à partir d’une loi de Bernoulli de probabilité  $p$ .

Après avoir entraîné les couches du réseau avec des couches de MonteCarlo Dropout, on fait passer une même entrée par  $T$  réseaux de neurones différents pendant la validation en conservant l’activation de ces couches de dropout.

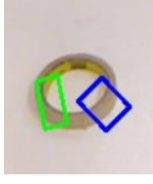
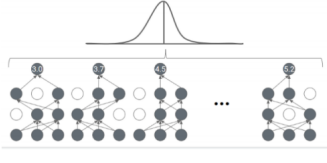
On note alors la sortie aléatoire du réseau  $f^W(x)$ . Dans le cas du Grasping, nous sommes dans le cas de la regression, et notre incertitude est calculée de la manière suivante pour une entrée de validation  $x^*$  :

$$E[y^*] \sim \frac{1}{T} \sum_{t=1}^T f^w_t(x^*) \quad (4)$$

$$V[y^*] \sim \frac{1}{T} \sum_{t=1}^T f^w_t(x^*)^T f^w_t(x^*) \quad (5)$$

Comme Srivastava et al. [2014], on choisit d’appliquer 50 masques de drop out pour les 100 données de validation comme illustré dans le tableau ci-dessous :

TABLE 5 – Méthodes d’évaluation des incertitudes epistémiques

Données d’entrée	50 masques de dropout	Résultats
		8 écarts-types

On adapte le calcul de l’incertitude proposé dans l’équation (5) à la problématique du Grasping comme détaillé dans l’équation (6) et (7).

$$\sigma_{xi} = \sqrt{\frac{1}{T} \sum_{t=1}^T (x_{ti} - \bar{x}_{ti})^2}$$

(6)

$$\sigma_{yi} = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_{ti} - \bar{y}_{ti})^2}$$

(7)

avec i compris entre 0;3, t compris entre 1;50

## 5 Résultats et expériences

### 5.1 Précision du réseau

Dans cette partie sont évaluées les performances de l’architecture de Redmon and Angelova [2015] avec une regression en apprentissage supervisé sur la base de données de l’université de Cornell. On garde 100 images de validation pour tester les performances du réseau.


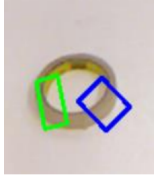
Après entrainement du réseau, la phase de validation sur les 100 images permet d’obtenir une précision du réseau de 80%. Cette précision est donc 10% plus faible que celle l’article de Redmon and Angelova [2015]. Pour rappel, la précision correspond à un ”bon” Grasping et un ”bon Grasping” correspond à un critère de Jacquard respecté.

Dans les 20% d’erreurs en phase de validation donc sur 20 images , on peut retrouver 12 images qui sont des faux négatifs. Le critère de Jacquard est trop restrictif car comme illustré sur la figure 6, l’objet aurait très bien pu être attrapé sur cette zone. Les 10 images restantes sont des vrais négatifs. Comme illustré dans le tableau suivant, l’agrapheuse ne peut pas être attrapée sur cette branche car trop fine et risque la chute lors de sa préhension. En réalité, il reste 8 prédictions du réseau vraiment problématiques sur les 100 prédictions qu’il effectue en phase de validation. Ces prédictions peuvent mettre en péril l’environnement dans lequel le réseau de neurones effectue sa prédiction.

Comme le confirme l’article de Depierre et al. [2018], la métrique de Jacquard est parfois trop restrictive et peut générer des faux négatifs comme l’illustre la figure 7 suivante. En effet, le rouleau de scotch aurait pu être attrapé sur tout son périmètre.

Néanmoins, il reste des vrais positifs comme sur la figure 6 où l’agrapheuse aurait du être attrapée sur la zone bleue.

TABLE 6 – Performances du réseau

Précision	Nbre d'images de validation	Nbre mauvaises prédictions	
80%	100	Faux négatifs 12	Vrais Négatifs 8
		Exemple de faux négatif	Exemple de Vrai Négatif
			

## 5.2 Des performances du réseau au calcul de l'incertitude des prédictions du réseau

D'après Ghoshal and Tucker, il est suffisant d'appliquer 50 masques de drop out sur chacune des images d'entraînement et de validation. On aura pour chaque entrée de validation 50 prédictions de rectangle de Grasping.

Chacune des coordonnées des rectangles de Grasping donnera lieu à 50 prédictions. Sur la figure suivante, on peut observer plusieurs prédictions pour une même entrée obtenue à partir de plusieurs masques de dropout.

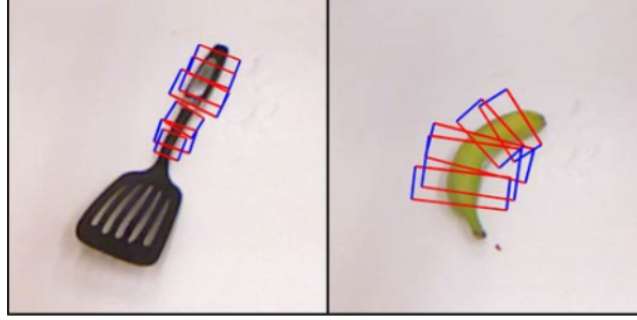


FIGURE 7 – Exemple de prédictions à partir d'un masque de dropout pour une même entrée

Pour chaque entrée, les 50 prédictions issues des 50 masques de drop out permettent de calculer l'incertitude du modèle. En effet, pour chaque entrée, 50 prédictions de rectangle de Grasping sont données par le réseau, ce qui permet de calculer une incertitude.

A partir des incertitudes obtenues pour les 100 entrées de validation, on calcule une moyenne des incertitudes sur chacune des coordonnées du rectangle de Grasping. Ces moyennes sont récapitulées dans le tableau 7 suivant :

TABLE 7 – Incertitude moyenne

$\sigma_{x0}$	$\sigma_{y0}$	$\sigma_{x1}$	$\sigma_{y1}$	$\sigma_{x2}$	$\sigma_{y2}$	$\sigma_{x3}$	$\sigma_{y3}$
$1.9.10^{-6}$	$1.3.10^{-5}$	$9,53.10^{-7}$	$7,62.10^{-6}$	$4,43.10^{-6}$	$8.28.10^{-6}$	$9.4.10^{-6}$	$10.4.10^{-6}$

L'entrée de validation qui a générée les incertitudes moyenne du tableau 7 est l'image 8 suivante :

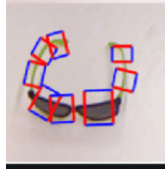


FIGURE 8 – Image correspondant à l’incertitude moyenne

L’écart observé entre les différents rectangles prédits est important. Comme l’explique Zheng et al. [2021], le seuil permettant de considérer qu’une prédiction d’un réseau est certaine ou non doit être fixé de manière empirique. Dans cet article, j’ai choisi de fixer ce seuil aux incertitudes moyennes du tableau 7.

On définit donc le seuil sur l’incertitude les valeurs suivantes pour chacune des coordonnées d’un rectangle de Grasping dans les deux tableaux suivants :

TABLE 8 – Seuil sur l’incertitude

Seuil sur $\sigma_{x0}$	Seuil sur $\sigma_{y0}$	Seuil sur $\sigma_{x1}$	Seuil sur $\sigma_{y1}$
$\sigma_{x0}^-$	$\sigma_{y0}^-$	$\sigma_{x1}^-$	$\sigma_{y1}^-$

TABLE 9 – Seuil sur l’incertitude

Seuil sur $\sigma_{x2}$	Seuil sur $\sigma_{y2}$	Seuil sur $\sigma_{x3}$	Seuil sur $\sigma_{y3}$
$\sigma_{x2}^-$	$\sigma_{y2}^-$	$\sigma_{x3}^-$	$\sigma_{y3}^-$

### 5.3 Du calcul de l’incertitude epistémique à la décision d’implémentation

A partir de ces écarts-types moyens on place un seuil empirique sur l’incertitude. La décision de faire confiance ou non à ce réseau réside selon Multaheb et al. [2020] dans la combinaison du critère de Jacquard et du seuil sur l’incertitude. Il précise que la matrice de confusion permet de visualiser très facilement les cas critiques.

TABLE 10 – Seuils de la matrice de confusion

	Prédiction certaine	Prédiction incertaine
Bonne prédiction	IoU>0.25 $\sigma_{x0} < \sigma_{x0}^- ; \sigma_{y0} < \sigma_{y0}^- ; \sigma_{x1} < \sigma_{x1}^- ; \sigma_{y1} < \sigma_{y1}^-$ $\sigma_{x2} < \sigma_{x2}^- ; \sigma_{y2} < \sigma_{y2}^- ; \sigma_{x3} < \sigma_{x3}^- ; \sigma_{y3} < \sigma_{y3}^-$	IoU>0.25 au moins une des des incertitudes supérieure aux seuils
Mauvaise prédiction	IoU<0.25 $\sigma_{x0} < \sigma_{x0}^- ; \sigma_{y0} < \sigma_{y0}^- ; \sigma_{x1} < \sigma_{x1}^- ; \sigma_{y1} < \sigma_{y1}^-$ $\sigma_{x2} < \sigma_{x2}^- ; \sigma_{y2} < \sigma_{y2}^- ; \sigma_{x3} < \sigma_{x3}^- ; \sigma_{y3} < \sigma_{y3}^-$	IoU<0.25 au moins une des incertitudes est supérieure aux seuils

Sur les images de validation, le réseau avec les méthodes d’évaluation des incertitudes effectue les prédictions suivantes :

Il reste donc 10% (10 images sur les 100 images de validation) mauvaises prédictions dont le réseau est certain.



TABLE 11 – Matrice de confusion

	Bonnes prédictions	Mauvaises prédictions
Certaines	60	10
Incertaines	20	10

## 6 Analyse critique des résultats

L’implémentation d’une étape supplémentaire permettant de calculer les risques inhérents aux prédictions des réseaux de neurones ne permet pas d’alerter sur toutes les erreurs du réseau et donc sur ses mauvaises prédictions. En effet, sur les 100 images de validation, le réseau s’est trompé 20 fois dont 10 où il est certain de sa prédiction même en y insérant une méthode d’évaluation des incertitudes. Si pour certains objets cela ne pose aucun problème, cela peut être dommageable pour d’autres.

TABLE 12 – Analyse comparative du % de prédictions fausses certaines de l’article à la littérature en utilisant le MC dropout comme méthode d’évaluation des incertitudes

Auteurs	Problème traité	Mauvaises prédictions certaines (%)
Kendall et al. [2016]	Segmentation d’image par apprentissage par apprentissage supervisée	7%
Abdar et al. [2021]	Classification de type de cancer de la peau par deep learning	2%
Aguilar and Radeva [2021]	Reconnaissance de nourriture	15%
Tanno et al. [2021]	Détection de tumeurs cérébrales sur des IRM	13%
Chai et al. [2020]	Détection de glaucomes sur des fonds d’oeils	20%
Balogopal et al. [2020]	Détection de cancer sur des radiothérapies	8%
Leibig et al. [2017]	Détection de rétinopathie sur des fonds d’oeils	15%
Ghoshal and Tucker	Détection de poumons atteint par la covid	7%
Vandal et al. [2018]	Prédiction de retard d’avions	5%

On peut en conclure même en ajoutant cette couche de sécurité supplémentaire, il ne paraît pas viable de mettre ce réseau de neurones sur une chaîne de production. Cette problématique de confiance dans les prédictions des réseaux de neurones se retrouvent dans d’autres domaines d’applications. Dans le tableau 12, pour chaque cas d’applications, l’évaluation des mauvaises prédictions certaines est très variable et reste très élevé. Cela est particulièrement dommageable lorsqu’il s’agit de détecter des tumeurs cérébrales sur des IRM ou des glaucomes sur des fonds ou il n’est pas possible de faire confiance à un réseau qui est certain de ses prédictions mêmes lorsque celles-ci sont fausses.

## 7 Conclusions

## Références

- Da Costa, Osaki, and Meads. Industry 4.0 in automated production. *Automation Control*, 2017.
- Bimont, Helenon, Nyiri, Thiery, and Gibaru. Easy grasping location learning from one-shot demonstration. *Arxiv*, 2020.
- Kleeberger, Bormann, Kraus, and Huber. A survey on learning-based robotic grasping. *Robotics in Manufacturing*, 2020.
- Schmidt, Vahrenkamp, Watcher, and Asfour. Grasping of unknown objects using deep convolutional neural networks based on depth images. *Arxiv*, 2020.
- Moon, Kim, Shin, and Hwang. Confidence-aware learning for deep neural networks. *arxiv*, 2020.
- Krizhevsky, Sutskever, and Hinton. Imagenet classification with deep convolutional neural networks. *arxiv*.
- Saxena, Driemeyer, and Ng. Robotic grasping of novel objects using vision. 2006.
- Caldera, Rassau, and Chai. Review of deep learning methods in robotic grasp detection. *Multimodal Technologies Interaction*, 2018.
- Kumra and Kanan. Robotic grasp detection using deep convolutional neural networks. *Arxiv*, 2017.
- Depierre, Dellandréa, and Chen. Jacquard : A large scale dataset for robotic grasp detection. *Arxiv*, 2018.
- Redmon and Angelova. Real-time grasp detection using convolutional neural networks. *arXiv*, 2015.
- Jiang, Moseson, and Saxena. Efficient grasping from rgb-d images : Learning using a new rectangle representation. *IEEE International Conference on Robotics and Automation*, 2015.
- Lenz, Lee, and Saxena. Deep learning for detecting robotic grasps. *arXiv*, 2015.
- Asif, Tang, and Harrer. Densely supervised grasp detector (dsgd). *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.
- Guo, Sun, Liu, Kong, Fang, and Xi. A hybrid deep architecture for robotic grasp detection. *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- Chu, Xu, and Vela. Real-world multi-object-multi-grasp detection. *IEEE Robotics and automation letters*, 2020.
- Gal. Uncertainty in deep learning. *Nature*, 2016.
- McKay. A practical bayesian framework for backpropagation networks. *Computation and Neural System California Institute of Technology*, 1992.
- Neal. Bayesian learning for neural networks. 1996.
- Graves. Practical variational inference for neural networks. 2011.
- Gal and Ghahramani. Dropout as a bayesian approximation : Representing model uncertainty in deep learning. 2019.
- Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

- Weytjen and Weerdt. Learning uncertainty with artificial neural networks for improved remaining time prediction of business processes. *LIRIS*, 2021.
- Kendall and Gal. What uncertainties do we need in bayesian deep learning for computer vision. *Journal of Machine Learning*, 2017.
- Shorten and Khoshgoftaar. A survey on image data augmentation for deep learning. *journal of Big Data*, 2015.
- Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning*, 2014.
- Ghoshal and Tucker. Uncertainty and interpretability in deep learning for coronavirus (covid-19) detection.
- Zheng, Zhang, Liu, and Sun. Uncertainty in bayesian deep label distribution learning. *Applied Soft Computing Journal*, 2021.
- Multaheb, Zimmering, and Niggemann. Expressing uncertainty in neural networks for production systems. *Automatisierungstechnik*, 2020.
- Kendall, Badrinarayanan, and Cipolla. Bayesian segnet : Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *University of Cambridge*, 2016.
- Abdar, Samani, Mahmoodabad, Doan, Mazoure, Hashemifesharaki, Liu, Khosravi, Acharya, Makarenkov, and Nahavandi. Uncertainty quantification in skin cancer classification using three-way decision-based bayesian deep learning. *Computers in Biology and Medicine*, 2021.
- Aguilar and Radeva. Deep learning and uncertainty modeling in visual food analysis. 2021.
- Tanno, Worral, Kaden, Ghosh, Grussu, Bizzi, Sotiripoulos, Criminisi, and Alexander. Uncertainty modelling in deep learning for safer neuroimage enhancement : Demonstration in diffusion mri. *NeuroImage*, 2021.
- Chai, Bian, Liu, Li, and Xu. Glaucoma diagnosis in the chines context : An uncertainty information-centric bayesian deep learning model. *Information Processing and Management*, 2020.
- Balagopal, Nguyen, Morgan, Weng, Dohopolski, Lin, Barkousaraie, Gonzalez, Garant, Desai, Hannan, and Jiang. A deep learning-based framework for segmenting invisible clinical target volumes with estimated uncertainties for post-operative prostate cancer radiotherapy. *Medical Image Analysis*, 2020.
- Leibig, Allken, Ayhan, Berens, and Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Nature*, 2017.
- Vandal, Livingston, Piho, and Zimmerman. Prediction and uncertainty quantification of daily airport flight delays. *Proceedings of Machine Learning Research, 4th International Conference on Predictive Application and APIs*, 2018.

## Table des figures

1	Définition d'une zone de Grasping . . . . .	6
2	Illustration du critère de Jacquard . . . . .	7
3	Exemple de résultats corrects prédits par un réseau de neurones . . . . .	7
4	Masque de Dropout . . . . .	10
5	Structure neuronale utilisée . . . . .	11
6	Explication de la constitution de la base de données de Cornell . . . . .	12
7	Exemple de prédictions à partir d'un masque de dropout pour une même entrée . . . . .	15
8	Image correspondant à l'incertitude moyenne . . . . .	16

## Liste des tableaux

1	Evolution des performances des structures neuronales appliquées à la problématique du Grasping . . . . .	8
2	Base de données de l'université de Cornell récupérée . . . . .	12
3	Répartition des données d'entraînement et de validations après Data Augmentation . . . . .	13
4	Paramètre d'entraînements du réseau . . . . .	13
5	Méthodes d'évaluation des incertitudes épistémiques . . . . .	14
6	Performances du réseau . . . . .	15
7	Incertainité moyenne . . . . .	15
8	Seuil sur l'incertainité . . . . .	16
9	Seuil sur l'incertainité . . . . .	16
10	Seuils de la matrice de confusion . . . . .	16
11	Matrice de confusion . . . . .	17
12	Analyse comparative du % de prédictions fausses certaines de l'article à la littérature en utilisant le MC dropout comme méthode d'évaluation des incertitudes . . . . .	17