

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
Elouanes Khelifi

RÈGLES ASSOCIATIVES FLOUES ET LOGIQUE DE RAISONNEMENT DANS
LES MODÈLES DE
VISION PAR ORDINATEUR (FUZZY YOLO-V5)

AVRIL 2023

REMERCIEMENTS

Tout d'abord, je suis profondément reconnaissant envers Dieu, qui m'a donné la force, la patience et la persévérance pour mener à bien ce projet. Sans sa grâce, rien n'aurait été possible. J'ai confiance que sa lumière continuera de me guider dans toutes mes futures entreprises.

Je suis également infiniment reconnaissant envers mes parents pour leur amour, leur soutien et leurs sacrifices inconditionnels tout au long de ma vie. Leur influence a été le fondement de mes aspirations et la force motrice qui m'a permis de persévérer dans les moments difficiles. Leur fierté et leur amour me donne la force de continuer à avancer.

Je tiens aussi à remercier mon professeur, le Dr Usef Faghihi, pour son mentorat, sa patience et son soutien indéfectible tout au long de ma thèse. Sa passion pour son sujet et son expertise ont été une source d'inspiration pour moi, et j'ai été honoré de travailler sous sa tutelle.

Je remercie également mes collègues, mes amies et mes partenaires de recherche pour leur collaboration, leur partage de connaissances et leur contribution à la réussite de ce projet. Leurs idées et leur expertise ont enrichi mon travail et m'ont permis d'élargir mes horizons.

Enfin, je suis reconnaissant envers toutes les personnes qui ont participé de près ou de loin à la réalisation de ce projet. Leur engagement et leur contribution ont été essentiels à sa réussite, et je suis fier de pouvoir partager le fruit de notre travail ensemble.

Je suis convaincu que cette expérience m'a apporté de précieuses leçons et m'a permis de développer des compétences qui me seront utiles tout au long de ma vie professionnelle. Je suis impatient de continuer à apprendre et à contribuer à la recherche dans mon domaine.

DEDICACE

À mon père,

Je dédie cette thèse à toi, qui a été mon plus grand soutien tout au long de mon parcours académique. Je suis tellement reconnaissant pour l'amour, le support et l'encouragement que tu m'as toujours apporté. Ton dévouement envers l'éducation et l'apprentissage m'a inspiré à persévérer malgré les obstacles.

Tu m'as appris l'importance du travail acharné, de la persévérance et de l'humilité. Tu as été un exemple de détermination et d'intégrité pour moi, et je suis fier d'être ton fils. Ta présence inconditionnelle dans ma vie a été un cadeau inestimable, et je ne pourrais jamais te remercier assez pour tout ce que tu as fait pour moi.

Je voudrais également profiter de cette occasion pour remercier dieu de t'avoir aidé vaincre le cancer et de rester avec nous pour voir ce jour. Ta force et ta résilience face à cette épreuve ont été une source d'inspiration pour moi, et j'admire ton courage et ta détermination.

Je suis énormément reconnaissant pour chaque jour que nous avons passé ensemble, et je suis fier de voir combien tu as accompli malgré les défis auxquels tu as été confronté. Ton optimisme et ta positivité ont été un phare de lumière pour notre famille pendant les moments difficiles, et je ne pourrais jamais te remercier assez pour tout ce que tu as fait pour nous.

Je suis heureux que tu sois ici pour célébrer cette réalisation avec moi, et je sais que tu continueras à être une source de soutien et d'inspiration pour moi à l'avenir. Je t'aime plus que tout au monde, et je suis fier de t'avoir comme mon père.

Cette thèse est un témoignage de l'influence positive que tu as eue sur ma vie. Je suis convaincu que ta sagesse, ton intelligence et ton amour ont été les fondements de ma réussite académique. J'espère que cette modeste réalisation te remplira de fierté et de joie, car c'est **grâce à toi** que j'en suis arrivé là.

Je t'aime de tout mon cœur, et je te dédie ce mémoire avec une profonde gratitude et une immense affection.

Ton fils dévoué.

TABLE DES MATIÈRES

TABLE DES FIGURES	v
ACRONYMES	vi
RÉSUMÉ	vii
CHAPITRE 1 INTRODUCTION	1
CHAPITRE 2 ETAT DE L'ART	2
2.1 ResNet	2
2.2 Attention Multi-Tête	3
2.3 U-net	4
2.4 YOLO.....	5
2.5 Identique différent (Same-Different)	6
2.6 BEiT	7
2.7 Phyre/DADS	7
CHAPITRE 3 FUZZY-YOLO-V5	8
3.1 CIOU flou (FIoU)	9
CHAPITRE 4 RAISONNEMENT AVEC YOLO-V5	14
CHAPITRE 5 CONCLUSION	19

TABLE DES FIGURES

Figure 2.1	Bloc résiduelle Resnet	2
Figure 2.2	Graph computationnel du mécanisme d'attention multi- têtes	3
Figure 2.3	Architecture U net.....	4
Figure 2.4	Architecture YOLO originale.....	5
Figure 3.1	Fonctions d'appartenance pour 5 niveaux de distance.....	10
Figure 3.2	Processus de calcul FIoU	10
Figure 3.3	Résultats comparatifs des mesures : Précision moyenne 0.5 et 0.95 (normes) pour YOLO avec CIoU et FIoU	12
Figure 3.4	Précision moyenne pour CIoU, FIoU avec règles générées automatiquement (FIoU-gen-rules), et FIoU avec règles définies par l'expert du domaine (FIoU-def-rules).....	13
Figure 4.1	Prédiction originale de YOLO-V5 sur des images recadrées de la même base de données.	14
Figure 4.2	Matrice de confusion de l'original YOLO-V5 sur des images recadrées.	15
Figure 4.3	Fonctions d'appartenance floues pour les prédictions concernant les phares.....	16
Figure 4.4	Calcul de la probabilité de classification par la logique de raisonnement.	16
Figure 4.5	La sortie défuzzifiée est la coordonnée x du centroïde de la zone jaune.....	17
Figure 4.6	Diagramme à barres comparatif de la précision de la classification des voitures entre Fuzzy YOLO-V5 et YOLO-V5 original formé sur Cars.	18

ACRONYMES

DL apprentissage profond (deep learning).

YOLO *You Only Look Once* (Vous ne regardez qu'une fois).

CEVAE Autoencodeurs variationnels à effet causal (*Causal Effect Variational Autoencoders*).

CIoU Intersection complète sur l'union.

IoU Intersection sur l'Union.

RESNETs Réseaux de neurones résiduels.

DIoU Distance Intersection sur Union (*Distance Intersection Over Union*).

GIoU Intersection sur Union Generale (*General Intersection Over Union*).

FIoU Intersection Floue sur l'union.

FAR règles d'association floues.

NMS *Non-Max Suppression*.

RL Apprentissage par renforcement.

RÉSUMÉ

De nombreux chercheurs ont pour objectif de résoudre des tâches humaines à l'aide d'algorithmes d'apprentissage automatique.

Dans ce mémoire, nous avons doté YOLOV-5 à des règles de logique floue qui peuvent être utilisées pour résoudre des problèmes de raisonnement en matière de classification (comme la classification ascendante) et de feedbacks linguistiques. Notre méthode utilise des règles d'association floues qui permettent à YOLOV-5 de déduire des connaissances à partir d'une vue partielle d'un objet et peuvent également être utilisées pour évaluer la perte de données à partir d'évaluations floues.

Mots-clés : Apprentissage profond, logique floue probabiliste, raisonnement, vision.

CHAPITRE 1

INTRODUCTION

Les algorithmes d'apprentissage profond (DL) deviennent extrêmement précis dans de nombreuses recherches et tâches industrielles liées à la détection d'objets [31], comme Gato de Google [26] et GPT d'OpenAI [5]. Parmi les différents DLs, You Only Look Once (**YOLO**) est un modèle très populaire connu pour sa rapidité et sa précision dans la détection d'objets [25]. YOLO et ses différentes versions (YOLO-V) ont été utilisés par des chercheurs et des ingénieurs pour résoudre des problèmes étroitement liés aux tâches humaines dans des laboratoires et des environnements industriels [41]. Cependant, les DLs actuels ne peuvent pas effectuer de raisonnement à la manière des humains [24]. En utilisant le modèle causal de Pearl [12], de nombreux chercheurs ont essayé d'ajouter le raisonnement causal aux algorithmes d'apprentissage profond (Deep Learning) [12, 19, 4, 22]. Cependant, à notre connaissance, le modèle de Pearl ne décrit dans la majorité des cas que la relation linéaire entre les variables et, dans certains cas, ne peut pas refléter la causalité [11]. En utilisant la logique floue probabiliste, Faghihi et al [9] ont ajouté le raisonnement causal aux Autoencodeurs Variationnels à effet causal (CEVAE) [21]. Toutefois, leur modèle ne peut raisonner que sur des ensembles de données tabulaires.

Dans ce travail, nous proposons la première étape vers la création de DL hybrides capables de raisonner de manière visuelle. Nous examinons également 1) comment un YOLOV doté de règles de logique floue peut concurrencer la formule Complete Intersection Over Union (CIOU) [16] utilisée depuis des années dans YOLO pour la détection d'objets. D'une manière générale, IoU et CIOU sont des mesures utilisées pour calculer les erreurs entre la boîte de délimitation correcte et prédite d'un objet détecté par un modèle tel que YOLO, 2) Comment un modèle YOLO équipé de règles de logique floue peut effectuer un raisonnement visuel. Ainsi, lorsqu'il est formé à la détection de certains objets, YOLO n'est pas en mesure d'identifier les images rognées de ces objets s'il n'a pas été préalablement formé sur ces rognures. Cependant, un être humain peut identifier correctement un objet, par exemple une voiture, en ne regardant que les différentes parties de l'objet, même s'il n'en a qu'une vision partielle.

En utilisant des règles d'association floues, nous proposons une technique généralisable qui peut aider à étendre les réseaux de neurones avec des capacités de raisonnement simples. Ensuite, nous discutons les résultats obtenus avec Fuzzy-YOLO et nous comparons avec les résultats du modèle yolo de référence.

CHAPITRE 2 ETAT DE L'ART

Le périmètre de ce travail ne permet pas d'effectuer un examen exhaustif des algorithmes de Deep Learning développés pour la détection d'objets. Cependant, nous aborderons brièvement certains d'entre eux dans la suite de ce mémoire. Toutefois, nous allons brièvement explorer certaines des approches les plus courantes utilisées pour la détection d'objets à partir de données visuelles. L'objectif est de fournir une vision d'ensemble des techniques et des défis liés à ce domaine de recherche en pleine évolution.

2.1 ResNet

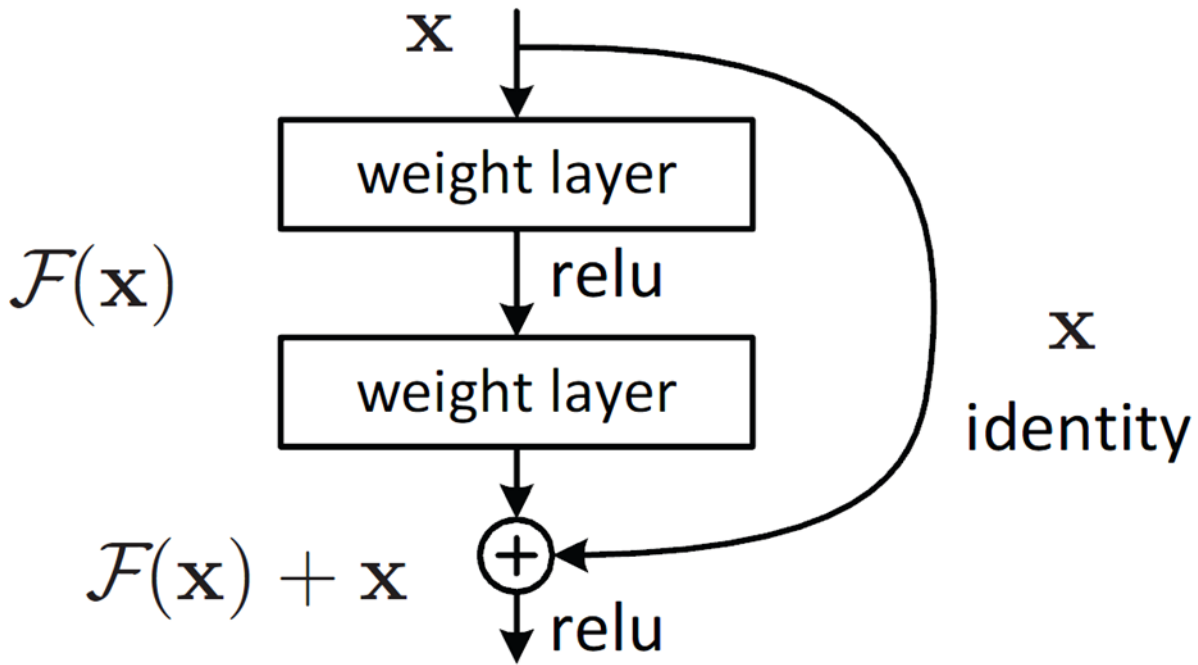


FIGURE 2.1 Bloc résiduelle Resnet Figure tirée de [14]

Le nombre de couches cachées dans les DLs joue un rôle crucial dans leurs performances. Cependant, plus la profondeur est grande, plus la précision du réseau se dégrade. Pour résoudre ce problème, les auteurs [14] ont introduit les réseaux de neurones résiduels (ResNet). Le ResNet comporte de nombreuses couches pondérées. Une fonction d'activation dérivable, telle que ReLU, est appliquée à la sortie de chaque couche pondérée. La sortie finale des couches est ensuite soustraite de l'entrée originale. Le réseau utilise ensuite la rétropropagation pour réduire les différences entre la sortie et l'entrée du réseau. Cela garantit l'intégrité de

l'information grâce à la profondeur des couches et des convolutions.

2.2 Attention Multi-Tête

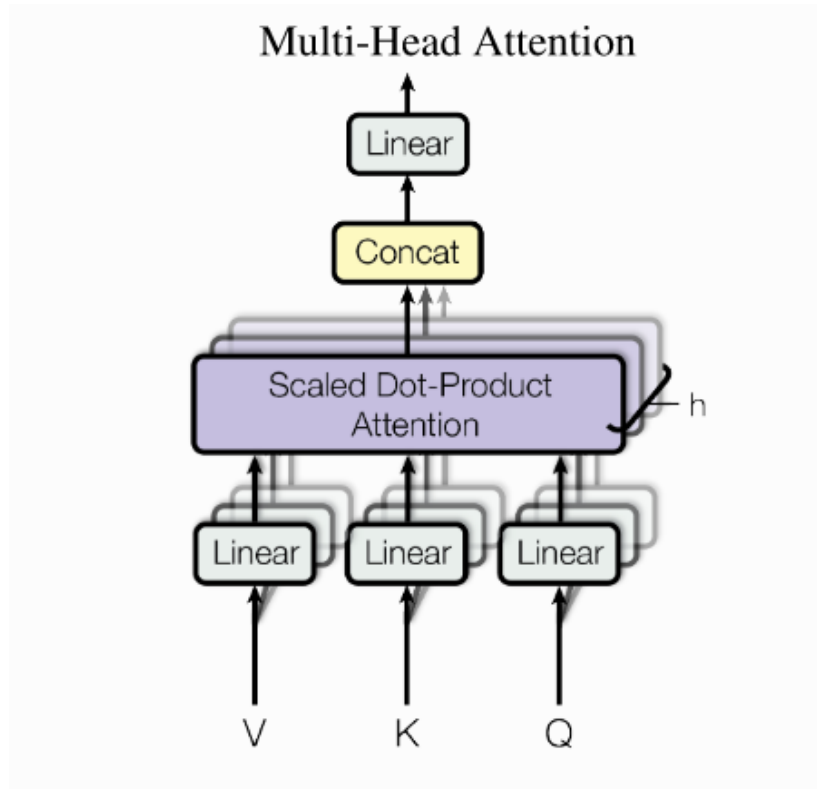


FIGURE 2.2 Graph computationnel du mécanisme d'attention multi-têtes (Multi-Head attention) figure tirée de [35]

Bien qu'il soit utilisé pour le traitement d'images, le mécanisme d'auto-attention a été adopté dans le domaine du traitement du langage naturel [44] où les mots sont intégrés à l'aide de vecteurs. Les vecteurs représentent la signification des mots sans tenir compte du contexte. Pour coder le contexte d'une phrase donnée, l'auto-attention intègre des mots en comparant chaque mot de la phrase avec tous les mots de la phrase (auto-inclus) [30]. Dans une définition plus abstraite, un mécanisme d'auto-attention est un mappage d'une requête et d'un ensemble de paires clé-valeur vers un résultat sous forme de vecteurs. Pour étendre ce concept à un contexte encore plus large, un mécanisme d'attention multi-têtes génère une description précise des mots en tenant compte des multiples facteurs qui peuvent affecter leur signification [35].

L'attention est traduite en réseaux de neurones convolutionnels pour la reconnaissance d'images, par des couches qui effectuent l'agrégation et la transformation des caractéristiques. Les fonctions d'agrégation et de transformation peuvent être découplées, de sorte que la transformation des caractéristiques est effectuée par

des couches de perception. Les couches de perception traitent séparément le vecteur de caractéristiques de chaque pixel. La fonction d'agrégation des caractéristiques est exécutée par auto-attention pour sélectionner les pixels les plus importants [44].

2.3 U-net

Basé sur des "réseaux entièrement convolutifs", U-net est une architecture conçue pour la segmentation sémantique. Par rapport à d'autres DL, U-net a été conçu pour fonctionner avec peu d'images d'entraînement. L'architecture U-net se compose de deux parties principales : la partie contractante qui est typiquement constituée de réseaux convolutifs, et la partie expansive qui est un échantillonnage ascendant de la carte des caractéristiques suivi d'une convolution 2x2 [28]. En plus de la classification des images, U-net localise les caractéristiques de chaque classe générée en effectuant la classification sur chaque pixel [43].

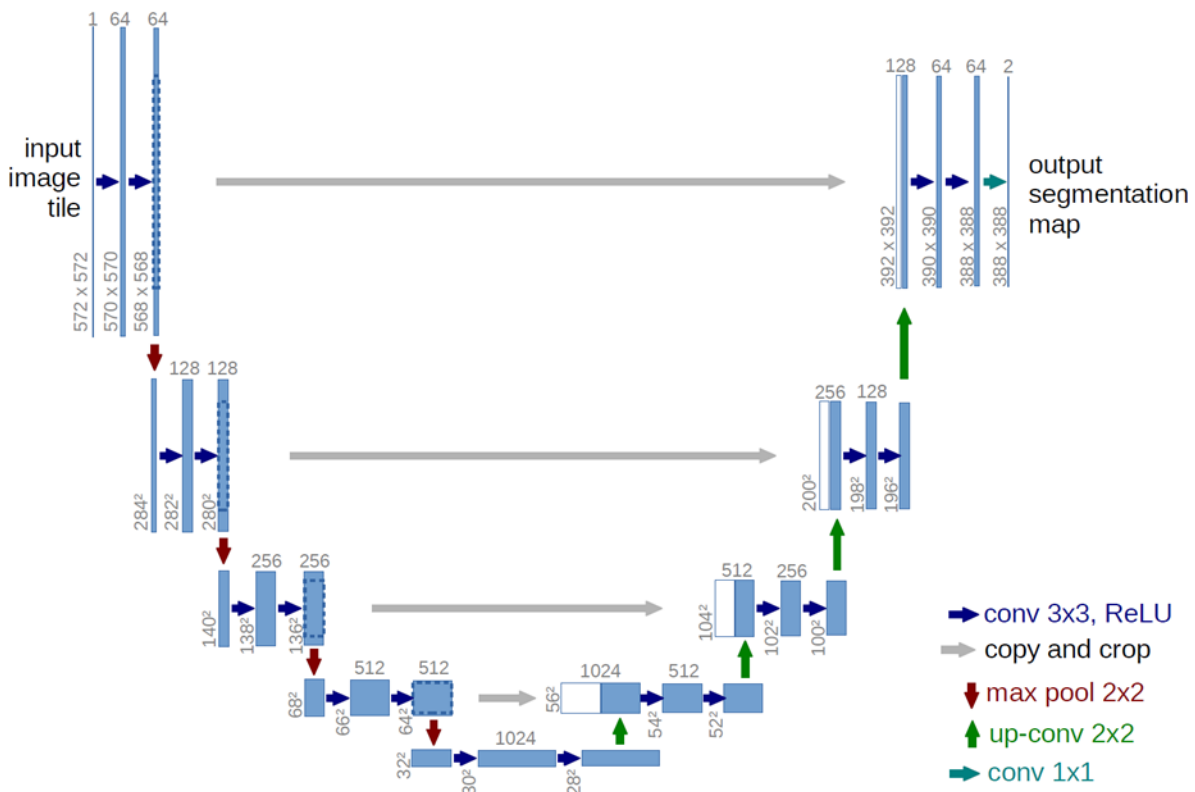


FIGURE 2.3 Architecture U net figure tirée de [28]

Dans certains cas, les auteurs utilisent ResNet en combinaison avec Unet, car il a été démontré que cela améliorerait la précision du modèle [6]. Pour mettre en œuvre cette combinaison, des blocs résiduels peuvent

être ajoutés à certaines zones du modèle Unet classique. Les blocs ResNet sont souvent ajoutés en tant qu'encodeur (sur le côté gauche de l'architecture Unet) pour pouvoir aller plus loin dans les convolutions et améliorer les résultats de la segmentation en termes de précision [6, 34, 2, 15]. Cependant, Unet est lent par rapport à YOLO et n'est utile que lorsque la segmentation est nécessaire.

2.4 YOLO

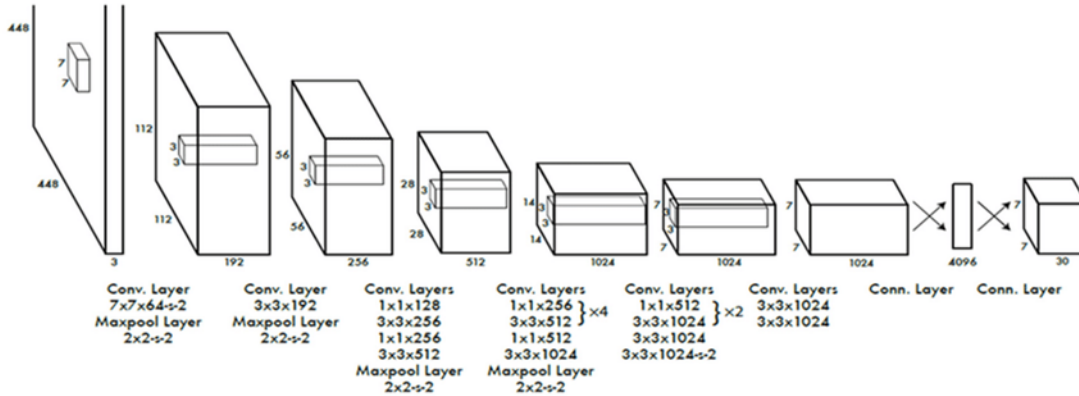


FIGURE 2.4 Architecture YOLO originale figure tirées de [25]

L'architecture appelée YOLO pour "You Only Look Once" (Vous ne regardez qu'une fois) prend en entrée des images étiquetées avec des vecteurs qui définissent toutes les informations nécessaires pour générer des boîtes de délimitation autour des objets détectés. Dans un problème de classification où il existe deux classes d'objets, ces vecteurs prendraient la forme $V \left[P_c, B_x, B_y, B_w, B_h, C_1, C_2 \right]$. Dans le vecteur V , $[P_c]$ est la probabilité de détecter un objet. $[B_x]$ et $[B_y]$ sont les coordonnées des centres de la boîte englobante autour de l'objet détecté par l'algorithme [25]. $[B_w]$ et $[B_h]$ sont la hauteur et la largeur de la boîte englobante. $[C_1]$ et $[C_2]$ sont la probabilité que l'objet détecté appartienne respectivement à la classe un ou à la classe deux. Chaque image est divisée en plusieurs cellules qui représentent les régions à rechercher. Chaque cellule prédit ces boîtes de délimitation. Trois types d'erreurs doivent être calculés séparément : 1) la perte de régression des boîtes de délimitation calcule l'intersection sur l'union (IoU), qui est la fraction de la zone d'intersection entre la zone de la boîte de délimitation cible et la zone de la prédiction du modèle sur leur union, 2) la perte d'objectivité calcule la probabilité d'avoir un objet détectable dans la boîte de délimitation prédite, 3) la perte de classification, qui est obtenue par l'entropie croisée¹. La perte totale est la somme moyenne des trois erreurs. Cependant, le principal défaut de la mesure de l'indice d'utilité ci-dessus est

1. L'entropie croisée est une mesure de la divergence entre deux distributions de probabilité.

qu'aux premiers stades du processus d'apprentissage, l'indice d'utilité peut toujours être égal à zéro, que les prédictions du modèle se rapprochent ou non de la prédiction correcte. Pour remédier à cet inconvénient, dans les versions récentes de YOLO, les chercheurs ont utilisé l'intersection complète sur l'union (CIoU), qui prend en compte l'intersection sur l'union, la distance normalisée entre les centres, et la cohérence du rapport d'aspect entre la boîte de délimitation de la cible et la prédiction du modèle. Par rapport à la version précédente, les techniques ci-dessus ont permis de faire converger la perte de YOLO beaucoup plus rapidement [16].

D'autres mesures qui résolvent le problème de la valeur zéro dans les premières étapes de l'apprentissage de YOLOV sont l'indice d'utilité de la distance (DIoU) et l'indice d'utilité généralisé (GIoU). Le DIoU prend en compte la distance entre les centres des boîtes de délimitation. Le GIoU prend en compte la plus petite boîte couvrant à la fois la cible et la prédiction du modèle [27]. Chacune de ces deux valeurs est ajoutée à la perte en tant que terme de pénalité, ce qui oblige le réseau à mettre à jour les poids de manière à réduire la pénalité par le biais de la descente de gradient.

Dans tous les modèles susmentionnés, l'élément manquant est le raisonnement. Les humains peuvent facilement effectuer des tâches de reconnaissance parce qu'ils utilisent le raisonnement en même temps que la perception pour déterminer la nature des objets qu'ils perçoivent. Il existe de nombreuses façons de définir une tâche de raisonnement. Dans le contexte de la réponse à des questions, une tâche de raisonnement consisterait à répondre à des questions telles que : "Dans une image donnée, quel objet se trouve derrière l'objet de couleur verte ?" [40]. Nous pensons que les règles d'association floues [10] peuvent également répondre à ce type de questions, comme nous l'expliquons plus loin dans ce document. Dans les sous-sections suivantes (2.5 , 2.6 , 2.7), nous discutons des modèles et des références qui utilisent le raisonnement.

2.5 Identique différent (Same-Different)

Dans un article récent, Nicola Messina et al [23] ont ajouté le mécanisme d'auto-attention à une architecture de réseau de neurones convolutionnel pour résoudre les tâches de raisonnement "identique-différent". Ces tâches impliquent la compréhension de règles abstraites à partir des formes de l'objet détecté. Leur architecture est plus performante que d'autres modèles pour résoudre ces types spécifiques de tâches de réponse aux questions dans la base de données " identique-différent". Cependant, leur modèle n'est pas capable de se généraliser à d'autres tâches telles que la classification, comme notre modèle implémenté peut le faire.

2.6 BEiT

Le modèle BEiT de Microsoft [38] est un modèle semi-général qui peut être utilisé pour répondre à des tâches liées à l'image et au texte. Il peut également effectuer des tâches de classification et de segmentation d'images. Les auteurs ont testé le modèle sur diverses tâches de réponse à des questions visuelles telles que les réponses aux questions des ensembles de données NLVR2, VQAv2 et COCO-Captioning. BEiT utilise un transformateur multivoie [33] pour fusionner les informations de plusieurs modèles pré-entraînés sur les images, le texte et les paires image/texte. Mais ce modèle ne peut effectuer que des tâches de raisonnement sur les images et le texte. Nous visons à concevoir un pipeline qui peut être étendu à des tâches de raisonnement plus générales telles que la déduction d'informations à partir d'observations partielles.

2.7 Phyre/DADS

L'ensemble de données Phyre [3] est un benchmark de raisonnement pour la physique qui requiert une certaine capacité de raisonnement et de compréhension de la physique pour que les modèles puissent le résoudre. L'un des meilleurs modèles capables de réaliser cette tâche est le travail d'Eltayeb Ahmed et al. de Google [1]. Comme le suggère le nom de leur article, le modèle a effectué un raisonnement physique à l'aide de "modèles sensibles à la dynamique". Les modèles sensibles à la dynamique sont des modèles d'apprentissage par renforcement qui peuvent apprendre un modèle de leur environnement sans supervision. Ils y parviennent en maximisant l'information mutuelle entre la paire d'actions de l'état actuel de l'agent et l'état suivant. L'information mutuelle est une mesure qui quantifie la quantité d'informations que nous obtenons sur une variable lorsque nous connaissons l'état d'une autre variable. En maximisant cette quantité, ils entendent apprendre des actions (ou des compétences qui sont des séquences d'actions) dont les résultats sont prévisibles [32]. Mais cette conception ne fonctionne que pour les problèmes de type apprentissage par renforcement et, à notre connaissance, elle n'est pas adaptée aux tâches de réponse aux questions ou de classification.

Dans le chapitre suivant, nous proposons un moyen simple d'intégrer le raisonnement à l'aide de règles d'association floues à YOLO-V5 pour classer des objets sans avoir été formé à leur sujet.

Remarque : Notez que depuis la publication de notre travail, une version plus récente de YOLO est disponible, nommée YOLO-V8, qui a été implémentée après la fin de notre étude.

CHAPITRE 3

FUZZY-YOLO-V5

Dans cette partie de notre travail, nous avons ajouté une couche de règles d'association personnalisables qui peuvent être définies par l'expert du domaine ou générées automatiquement à partir de données d'entraînement selon l'approche suggérée par Wang et Mendel [36]. En utilisant des règles floues (voir ci-dessous), nous avons converti la fonction de perte de YOLO-V5 en une fonction de perte floue. Pour déduire cette fonction de perte floue, nous avons utilisé des métriques d'évaluation littérales dans le calcul de l'erreur de régression de la boîte de délimitation de YOLO-V5. Cet ajout est une alternative à la méthode Complete Intersection Over Union (CIOU), qui est abordée dans la sous-section ci-dessous. Nous appelons cette alternative floue FIoU.

Définition 3.1 *Logique Floue (Fuzzy Logic) : La logique floue est une branche de la logique qui permet de traiter des situations où les frontières entre les catégories sont floues, imprécises ou incertaines. Elle a été introduite par Zadeh en 1965 [42] et repose sur l'utilisation de degrés d'appartenance plutôt que sur une classification binaire.*

La logique floue a trouvé de nombreuses applications dans les domaines de la reconnaissance de formes, du contrôle de processus, de la prise de décision, de l'analyse de risques, de la médecine et de l'ingénierie, entre autres [17, 8].

Les principales opérations de la logique floue sont l'union, l'intersection et la complémentation floues, qui permettent de combiner des ensembles flous pour en former de nouveaux. Les règles de la logique floue sont exprimées en termes de variables linguistiques, qui sont traduites en termes de fonctions d'appartenance floues.

Les systèmes experts basés sur la logique floue, tels que les systèmes de contrôle de processus, utilisent des règles linguistiques pour effectuer des inférences floues et prendre des décisions en temps réel [37]. La logique floue est également utilisée en combinaison avec les réseaux de neurones pour améliorer leur capacité à traiter des données incertaines et imprécises [13].

En résumé, la logique floue est une approche puissante pour la modélisation de systèmes complexes et incertains, qui a trouvé de nombreuses applications dans de nombreux domaines différents.

Scikit-Fuzzy¹ est une bibliothèque Python pour les systèmes de logique floue. Elle fournit des outils pour les fonctions d'appartenance floues, les opérateurs logiques flous et les méthodes d'inférence pour faciliter le développement de systèmes de logique floue. Scikit-Fuzzy est open source et est largement utilisé dans les domaines de l'ingénierie, de la science des données et de l'intelligence artificielle pour résoudre des problèmes complexes qui impliquent de l'incertitude ou des données imprécises.

L'exemple de pourboire de Scikit-Fuzzy est un exemple couramment utilisé pour illustrer l'utilisation de la logique floue dans la prise de décision. Dans cet exemple, on suppose qu'un client est allé dans un restaurant et a reçu un service. Le client souhaite maintenant laisser un pourboire au serveur, mais il n'est pas sûr de la quantité à donner.

Pour résoudre ce problème, on utilise la logique floue pour déterminer le pourcentage de pourboire approprié en fonction de plusieurs variables d'entrée telles que la qualité de service, la qualité de la nourriture, et la température de la nourriture. La logique floue permet de modéliser ces variables d'entrée comme des fonctions d'appartenance floues et de définir des règles floues pour déterminer le pourcentage de pourboire approprié.

L'exemple de pourboire de Scikit-Fuzzy est souvent utilisé pour montrer comment la logique floue peut être utilisée pour résoudre des problèmes réels, en fournissant des résultats précis et faciles à interpréter.

3.1 CIOU flou (FIoU)

Notre FIoU est une version alternative du CIOU qui utilise des variables linguistiques telles que "bonne prédiction", "prédiction moyenne" et "mauvaise prédiction" pour calculer la perte (voir figure 3.1). Pour calculer la valeur de perte de la boîte de délimitation à l'aide de règles d'association floues (FARs), nous avons procédé en premier lieu à la fuzzification des données d'entrée. Pour ce faire, nous transformons les données d'entrée (distance normalisée, intersection sur l'union et cohérence du rapport d'aspect) de valeurs numériques à des valeurs linguistiques. Nous avons choisi cinq fonctions d'appartenance floues [7] pour

1. skfuzzy 0.2 docs (pythonhosted.org)

chaque entrée (figure 3.1) : très élevée, élevée, moyenne, faible ou très faible.

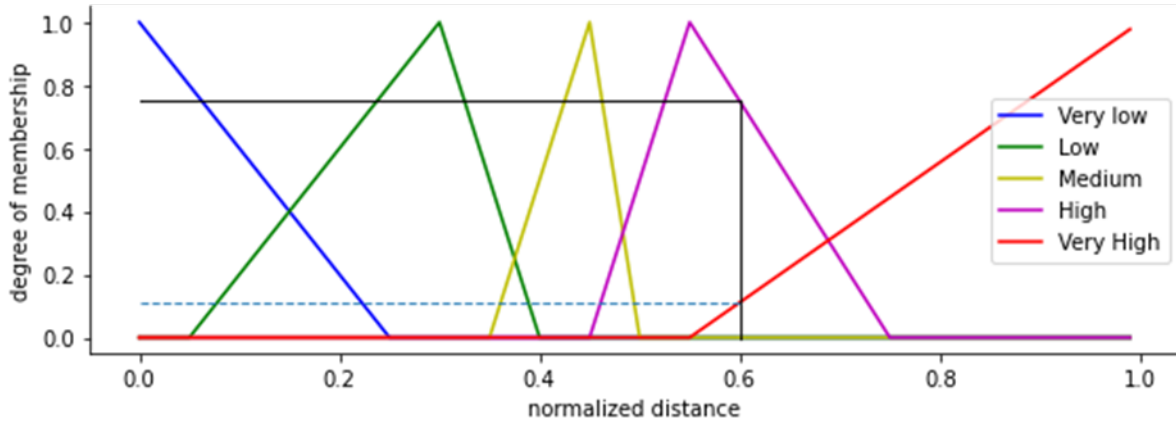


FIGURE 3.1 Fonctions d'appartenance pour 5 niveaux de distance. Cette figure montre que la valeur 0.6 pour la distance normalisée a une appartenance de 0.75 au niveau "élevé", de 0.17 au niveau "très élevé" et de 0 pour les trois autres niveaux.

après avoir flouté les métriques d'entrée (distance normalisée, intersection sur union et cohérence du rapport d'aspect) comme le montre la figure 3.2 (module Fuzzy Engine), nous avons appliqué des règles d'association floues aux métriques floutées de sortie. En appliquant ces règles, nous avons pu estimer la "qualité" de la prédiction à l'aide de la fonction de défuzzification. Dans notre expérience, nous avons nommé cette métrique : FIoU.

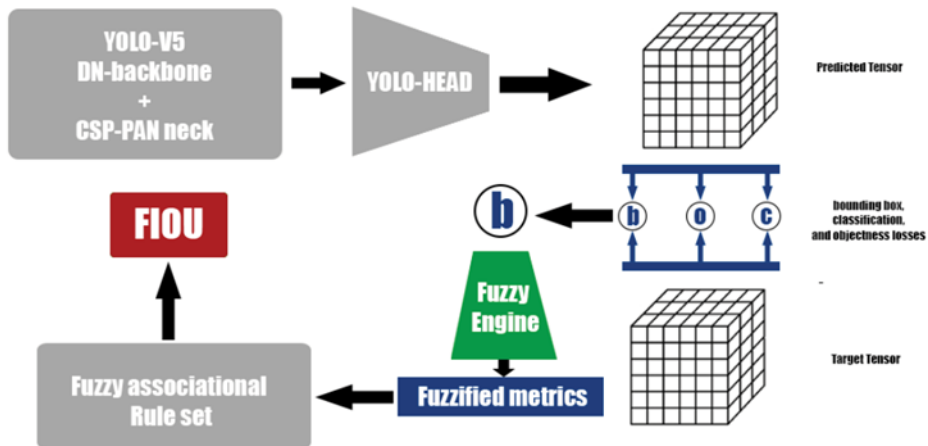


FIGURE 3.2 Processus de calcul FIoU : FIoU ne prend en compte que la perte de régression des boîtes englobantes indiquée par "b" dans ce diagramme. La perte d'objectivité (probabilité d'être un objet détectable) "o" et la perte de classification "c" ne sont pas modifiées. Dans l'original yolo-v5, 'b' est '1-CIoU' et dans notre travail '1-FIoU'.

Pour évaluer les performances de Fuzzy-Yolo-V5, nous avons utilisé les règles suivantes : X étant une variable d'entrée, et par exemple $X_{tfaible}$ le degré d'appartenance de X qui est assigné au niveau "très faible".

$$R_1 : D_{télévée} \nabla IoU_{tfaible} \nabla V_{tfaible} \Rightarrow FIoU_{tfaible}$$

$$R_2 : D_{élevée} \nabla (IoU_{faible} \Delta V_{faible}) \Rightarrow FIoU_{faible}$$

$$R_3 : D_{moyenne} \Delta IoU_{moyenne} \Delta V_{moyenne} \Rightarrow FIoU_{moyenne}$$

$$R_4 : D_{tfaible} \nabla IoU_{télévée} \nabla V_{télévée} \Rightarrow FIoU_{télévée}$$

$$R_5 : D_{faible} \Delta IoU_{élevée} \Delta V_{élevée} \Rightarrow FIoU_{élevée}$$

$$R_6 : D_{tfaible} \Delta IoU_{télévée} \Delta V_{télévée} \Rightarrow FIoU_{télévée}$$

D : distance normalisée

V : cohérence du rapport d'aspect

IoU : intersection sur l'union

Δ : opérateur ET flou qui est le minimum des deux valeurs

∇ : opérateur OU flou qui est le maximum des deux valeurs

Les règles peuvent être considérées comme des poids qui sont automatiquement ajoutés à chaque terme dans la fonction de perte de Yolo-V5. Un poids plus élevé signifie un terme plus important dans la fonction de perte. Les poids sont attribués par des règles floues. Une fois la valeur FIoU obtenue, elle remplace le terme CIOU dans la fonction de perte comme suit.

$$Perte\ de\ régression = 1 - FIoU$$

En utilisant les règles susmentionnées, nous avons obtenu approximativement la même précision moyenne (MAP) que le CIOU de Yolo [39]. La figure 3.3 montre la comparaison entre les résultats de FIOU et de CIOU. Pour faire cela, nous avons utilisé l'ensemble de données COCO [20] et la comparaison est effectuée à l'aide des mesures MAP standard, MAP 0.5 et MAP 0.95 après 150 itérations. Comme le montre la figure 3.3, CIOU et FIOU se comportent de manière similaire.

Nous avons également généré les règles automatiquement selon la méthode suggérée dans [36] (code sur GitHub). La figure 3.4 montre la différence de précision moyenne entre les résultats de YOLO-V5 utilisant des règles générées automatiquement [36], et nos règles définies par des experts.

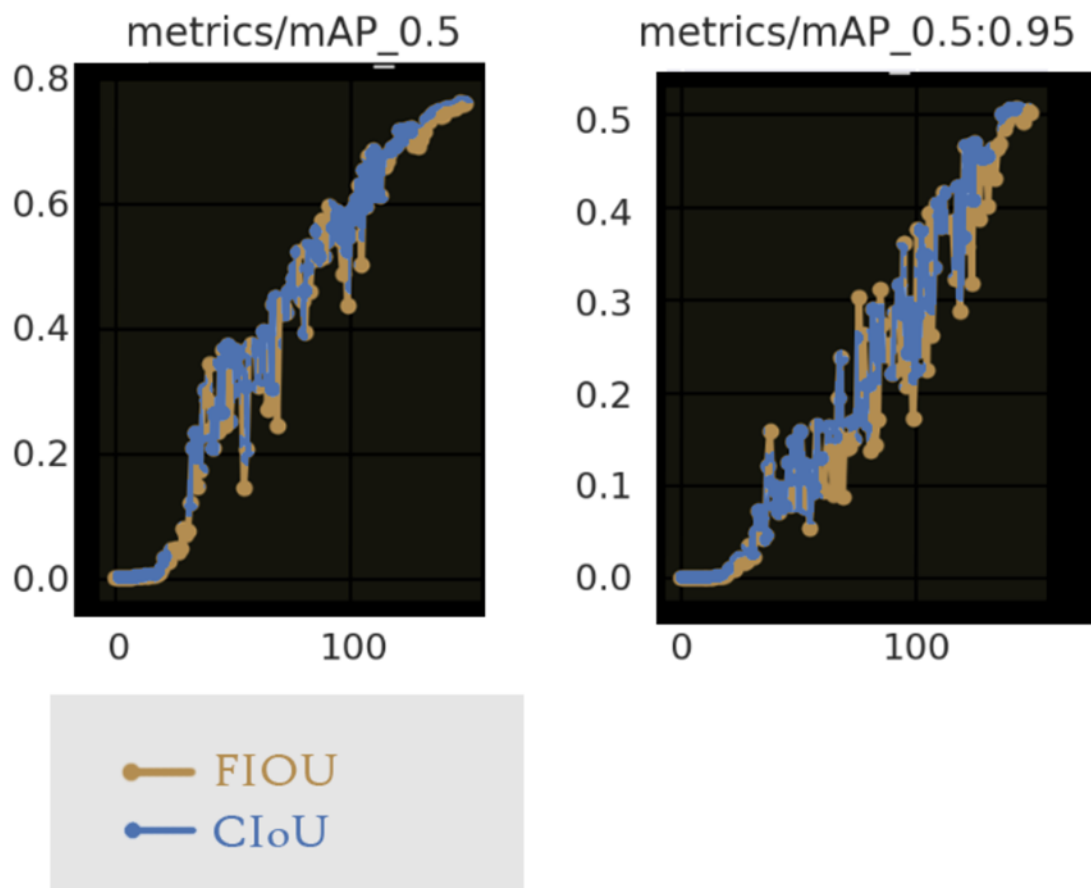


FIGURE 3.3 Résultats comparatifs des mesures : Précision moyenne 0.5 et 0.95 (normes) pour YOLO avec CIoU et FIOU

La figure 3.4 montre clairement que les règles d'experts (courbe noire) et les règles générées automatiquement (courbe violette) de [36] ont obtenu à peu près la même précision moyenne que CIoU.

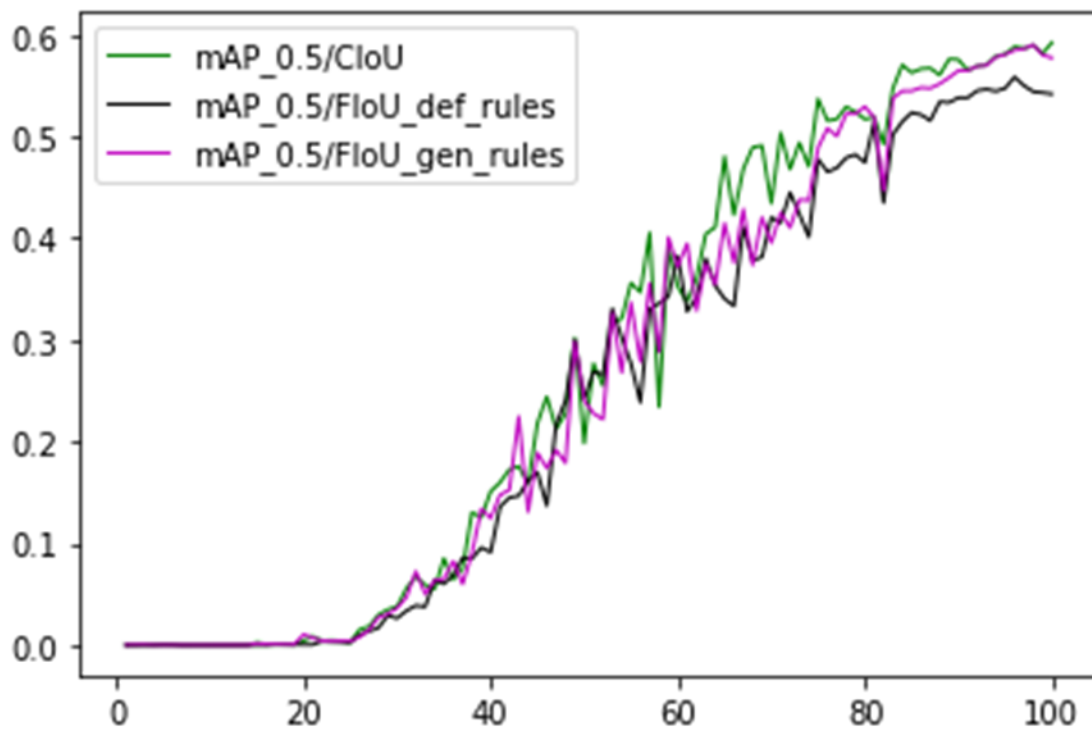


FIGURE 3.4 Précision moyenne pour Clou, FloU avec règles générées automatiquement (FloU-gen-rules), et FloU avec règles définies par l'expert du domaine (FloU-def-rules).

CHAPITRE 4

RAISONNEMENT AVEC YOLO-V5

Afin d'améliorer l'efficacité de YOLO-V5, nous expliquons comment les règles floues peuvent aider YOLO-V5 à effectuer une détection ascendante des objets. En ne voyant que les phares des voitures et d'autres parties en général, notre YOLO-V5 flou peut estimer la probabilité qu'un objet (partie d'une voiture) soit une voiture. Pour réaliser cette tâche, nous avons utilisé deux YOLO-V5 distincts formés sur la base de données CARS de Stanford [18] : 1) Notre modèle YOLO-V5 flou entraîné à détecter uniquement les parties d'une voiture telles que les phares, les rétroviseurs, les roues, le pare-brise et les feux de freinage, 2) Le modèle YOLO-V5 original entraîné à détecter les voitures entières mais pas les parties.

Nous avons testé les deux modèles YOLO-V5 sur des images de voitures recadrées. La figure 4.1 montre que le modèle original YOLO-V5 a des difficultés à détecter les voitures sur des images recadrées.



FIGURE 4.1 Prédiction originale de YOLO-V5 sur des images recadrées de la même base de données. YOLO-V5 ne détecte pas les voitures, et détecte à tort les phares comme des voitures.

Dans la figure 4.2, on peut voir que la précision de classification du YOLO-V5 original chute à 22 % lorsque les images sont recadrées. En d'autres termes, le YOLO-V5 original ne détecte parfois rien lorsque l'on zoome trop sur l'objet, s'il n'a pas été entraîné au préalable sur de telles images.

Dans YOLO-V5 flou, nous avons ajouté un moteur flou juste après le module Non-Max Suppression (NMS) (Figure 4.3). Comme Yolo divise l'image en cellules et que chaque cellule possède ses propres boîtes de

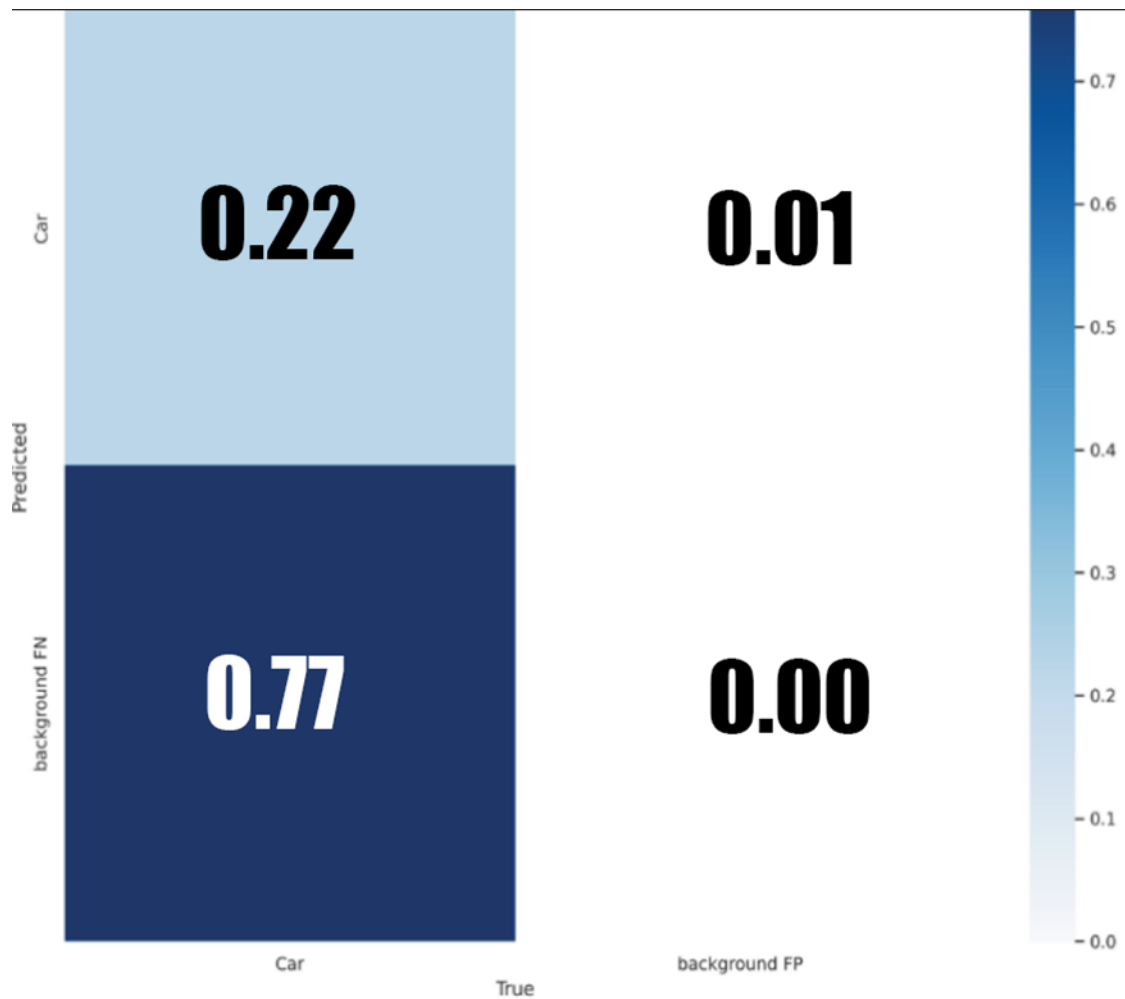


FIGURE 4.2 Matrice de confusion de l'original YOLO-V5 sur des images recadrées. La précision de la classification tombe à 22 %.

délimitation prédites, le module NMS est nécessaire pour éliminer les prédictions redondantes. Le module NMS effectue cette tâche en éliminant les boîtes d'encombrement qui décrivent très probablement le même objet, ne laissant qu'une seule boîte d'encombrement finale pour chaque objet présent dans l'image [29].

Le NMS produit des prédictions de classe "C" et un score de confiance "cS" (voir figure 3.3) qui sont ensuite transmis à un module de moteur flou. Le module Fuzzy Engine flouise ensuite ces valeurs sur trois niveaux (faible, moyen ou élevé). Enfin, il utilise des règles d'association floues et une fonction de défuzzification pour déduire la probabilité de la classe de l'objet final, par exemple une voiture.

En outre, pour calculer les degrés d'appartenance, nous avons choisi trois fourchettes (voir figure 4.4) : "faible, moyen ou élevé". Par exemple, si la tête YOLO détecte un phare avec une confiance de 0.7, cette

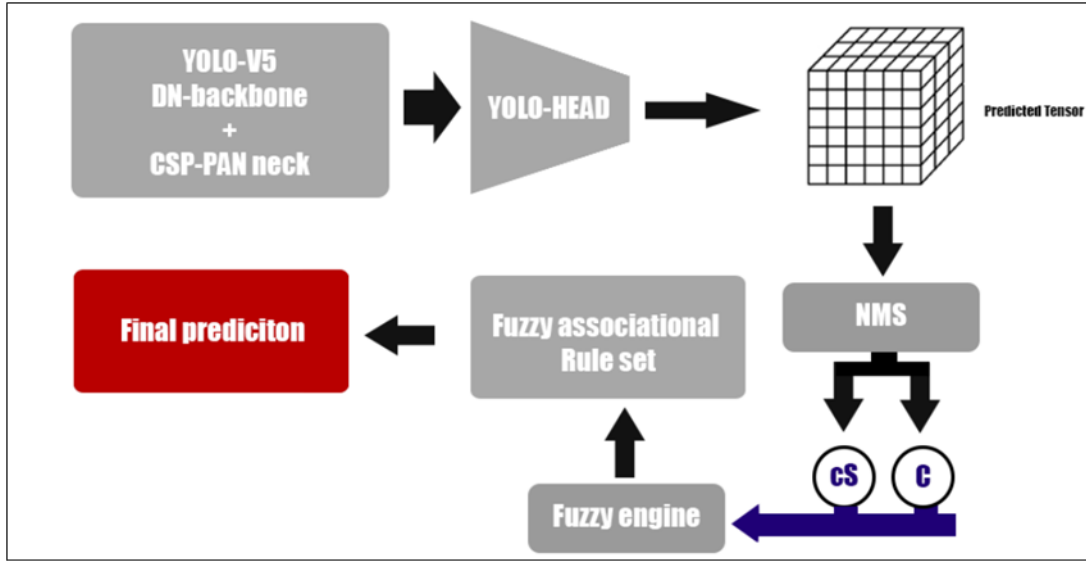


FIGURE 4.3 Fonctions d'appartenance floues pour les prédictions concernant les phares. Trois niveaux de certitude ont été définis pour déterminer le degré de confiance d'une prédiction.

valeur, une fois fuzzifiée, se verra attribuer une valeur d'appartenance de 0 pour les niveaux "moyen" et "faible" et une valeur d'appartenance de 0.6 pour le niveau "élevé". Par exemple, une sortie fuzzifiée pour un phare avec une appartenance de 0.6 signifie que les règles avec ($H_{élevée}$) seront activées par le moteur Fuzzy (voir les règles ci-dessous).

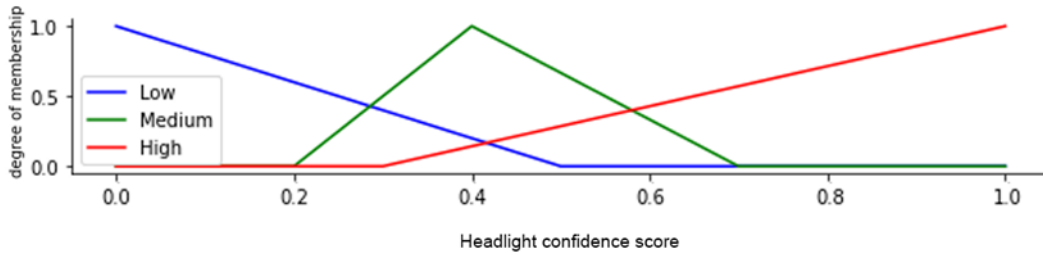


FIGURE 4.4 Calcul de la probabilité de classification par la logique de raisonnement.

Voici les règles que nous avons utilisées pour définir le raisonnement qui sous-tend la décision d'acheter ou de ne pas acheter une voiture par le moteur flou, en utilisant les mêmes annotations que celles utilisées dans les sections précédentes :

$$R_1 : (P_{élevé} \nabla Ff_{élevé}) \Delta (Pb_{élevé} \nabla Ré_{élevé} \nabla R_{élevé}) \Rightarrow C_{élevé}$$

$$R_2 : (P_{moyen} \nabla Ff_{moyen}) \Delta (Pb_{moyen} \nabla Ré_{moyen} \nabla R_{moyen}) \Rightarrow C_{élevé}$$

$$R_3 : (P_{faible} \Delta Ff_{faible}) \nabla (Pb_{faible} \Delta Ré_{faible} \Delta R_{faible}) \Rightarrow C_{faible}$$

$$R_4 : (P_{moyen} \nabla Ff_{moyen}) \Delta (Pb_{moyen} \Delta Ré_{moyen} \Delta R_{moyen}) \Rightarrow C_{moyen}$$

$$R_5 : (P_{faible} \nabla Ff_{faible}) \Delta (Pb_{faible} \Delta Ré_{faible} \Delta R_{faible}) \Rightarrow C_{moyen}$$

P : score de confiance des phares

Ff : score de confiance des feux de freinage

Pb : score de confiance du pare – brise

$Ré$: score de confiance de rétroviseur

R : score de confiance des roues

Enfin, la valeur de défuzzification peut être obtenue en calculant le centroïde de la zone jaune illustrée dans la figure 4.1. Selon la documentation du cadre flou de Sci-Kit, il existe quatre autres façons pour défuzzifier les données, mais c'est le centroïde qui a donné les meilleurs résultats.

Dans la figure 4.5, le score de confiance accordée au phare, pare-brise, rétroviseur, feu de freinage et à la roue est respectivement : 7.2, 2.0 et 7. Cela signifie que nous avons un score de confiance élevé pour le phare et la roue, et un score de confiance faible pour les autres pièces. Dans ce cas, conformément à la règle R1 (ci-dessus), le modèle doit attribuer une probabilité élevée au fait que l'objet est une voiture. La valeur estimée de la probabilité défuzzifiée pour la voiture est donc de 0.74.

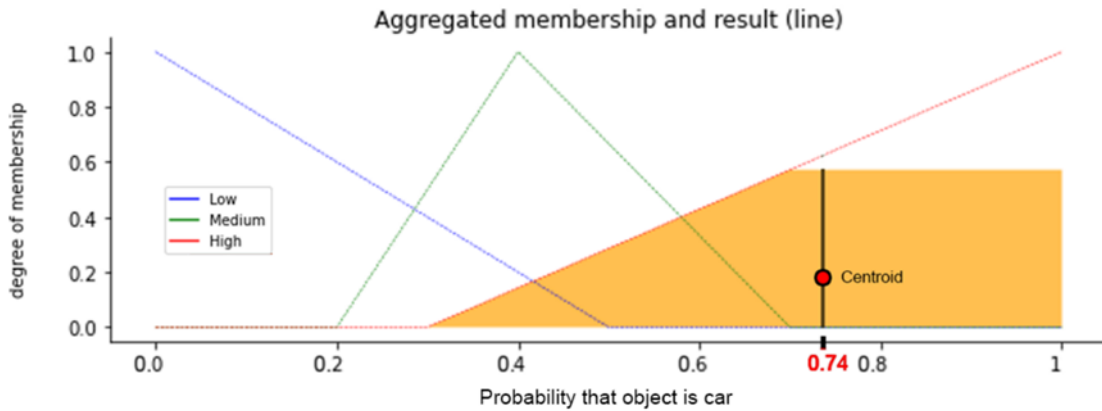


FIGURE 4.5 La sortie défuzzifiée est la coordonnée x du centroïde de la zone jaune.

Dans ce qui suit, nous comparons les résultats du YOLO-V5 original mentionnés ci-dessus à notre YOLO-V5 flou testé sur le même ensemble d'images recadrées extraites de l'ensemble de données Stanford cars [18]. Notre modèle considère que les probabilités de sortie des règles d'association floues dépassant la valeur seuil de 0.4 correspondent à une voiture. La figure 4.6 montre la précision de notre modèle YOLO-V5 flou

après l'utilisation des règles de raisonnement par rapport au modèle YOLO original formé à la détection des voitures.

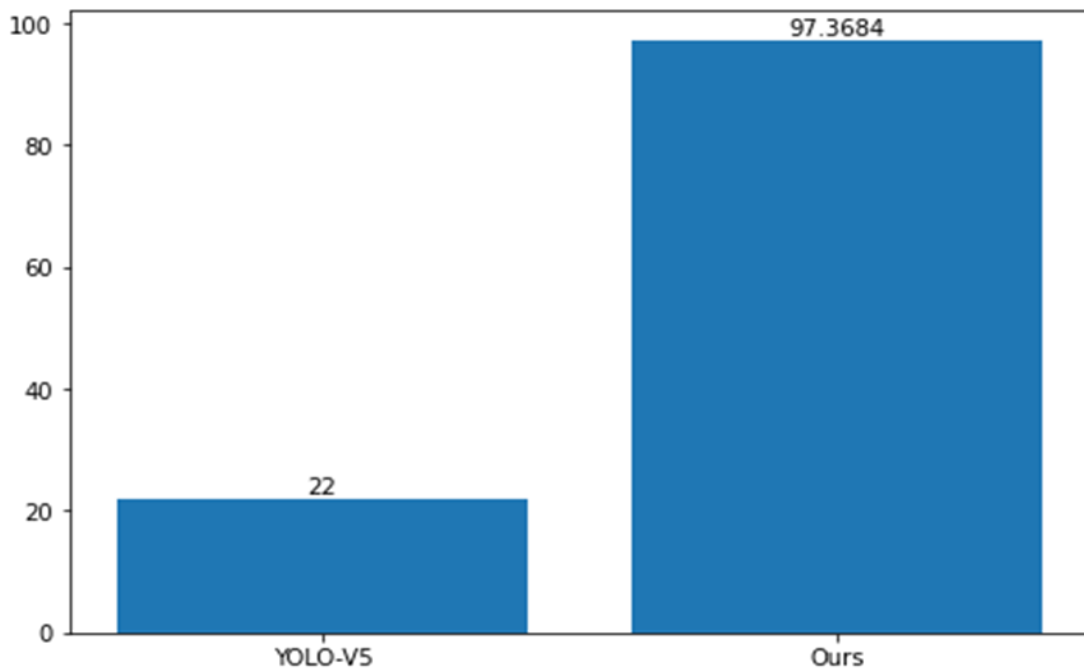


FIGURE 4.6 Diagramme à barres comparatif de la précision de la classification des voitures entre Fuzzy YOLO-V5 et YOLO-V5 original formé sur Cars. Les deux ont été testés sur des versions recadrées de l'ensemble de données Stanford sur les voitures.

Bien qu'il n'ait été entraîné que sur des pièces de voiture, notre modèle flou YOLO-V5 atteint une précision de 97 %, contre une précision de classification de 22 % obtenue par le modèle original YOLO-V5. Cela montre qu'avec cette technique, nous avons pu déduire correctement la probabilité que l'objet soit une voiture en utilisant des règles d'association floues pour simuler la détection ascendante d'objets et la tâche de classification.

Un point qui mérite d'être mentionné est que la portée de ce mémoire peut aller au-delà du modèle YOLO. Cela signifie que, conceptuellement, cette idée peut être étendue à n'importe quel algorithme d'apprentissage en profondeur tel que les transformateurs pour déduire de nouvelles informations. Sachant que les règles utilisées peuvent être apprises à l'aide de la technique de Wang et al [36] mentionnée dans les sections précédentes.

CHAPITRE 5

CONCLUSION

Dans ce document, nous proposons deux façons d'utiliser la logique floue pour doter les réseaux de neurones de capacités de raisonnement leur permettant de travailler en coopération avec les humains. Premièrement, nous avons montré que des règles d'association floues peuvent être ajoutées aux fonctions de perte de YOLO-V5 afin de simuler le CIoU flou sans perdre sa précision. Deuxièmement, les règles d'association floues peuvent également être utilisées comme extension du moteur de raisonnement pour permettre à YOLO-V5 d'effectuer un raisonnement ascendant. En utilisant ces règles floues, YOLO-V5 peut déduire de nouvelles connaissances sans avoir besoin d'être explicitement formé à celles-ci.

Dans nos travaux futurs, nous espérons que nos nouveaux moteurs flous aideront les algorithmes d'apprentissage automatique à apprendre plus rapidement avec moins d'itérations. Les règles floues mises en œuvre dans cette étude pourraient facilement être intégrées dans les fonctions de perte de l'apprentissage par renforcement (RL), ce qui permettrait aux humains de dire aux agents RL de corriger leurs erreurs en utilisant un langage vague au lieu de valeurs précises. Par exemple, le retour d'information humain pourrait être simplement "trop-loin" ou "trop-près" au lieu de "corrigez l'erreur de +9.9 mètres".

BIBLIOGRAPHIE

- [1] Eltayeb Ahmed, Anton Bakhtin, Laurens van der Maaten, and Rohit Girdhar. Physical reasoning using dynamics-aware models. *arXiv preprint arXiv :2102.10336*, 2021.
- [2] Yasser Ali, Farrokh Janabi-Sharifi, and Soosan Beheshti. Echocardiographic image segmentation using deep res-u network. *Biomedical Signal Processing and Control*, 64 :102248, 2021.
- [3] Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre : A new benchmark for physical reasoning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [4] Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. *arXiv preprint arXiv :2002.04083*, 2020.
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and Amanda Askell. Language models are few-shot learners. *Advances in neural information processing systems*, 33 :1877–1901, 2020.
- [6] Kaili Cao and Xiaoli Zhang. An improved res-unet model for tree species classification using airborne high-resolution images. *Remote Sensing*, 12(7) :1128, 2020.
- [7] Mu-Song Chen and Shinn-Wen Wang. Fuzzy clustering analysis for optimizing fuzzy membership functions. *Fuzzy sets and systems*, 103(2) :239–254, 1999.
- [8] S. M. Chen. methods for subjective mental workload assessment and fuzzy risk analysis. *Cybernetics Systems*, 1996.
- [9] Usef Faghihi, Cyrus Kalantarpour, and Amir Saki. Causal probabilistic based variational autoencoders capable of handling noisy inputs using fuzzy logic rules. In *Science and Information Conference*, pages 190–202. Springer.
- [10] Usef Faghihi, Serge Robert, Pierre Poirier, and Youssef Barkaoui. From association to reasoning. 2020.
- [11] Usef Faghihi and Amir Saki. Probabilistic variational causal effect as a new theory for causal reasoning. *arXiv preprint arXiv :2208.06269*, 2022.

- [12] Dennis Frauen and Stefan Feuerriegel. Estimating individual treatment effects under unobserved confounding using binary instruments. *arXiv preprint arXiv :2208.08544*, 2022.
- [13] S. Haykin and R. Lippmann. Neural networks a comprehensive foundation. *International journal of neural systems*, 1994.
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [15] Sagar Karki and Siddhivinayak Kulkarni. Ship detection and segmentation using unet. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–7. IEEE.
- [16] Renu Khandelwal. Different iou losses for faster and accurate object detection, 2021.
- [17] B. Kosko. *Fuzzy associative memory systems*. Fuzzy expert systems, 1992.
- [18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.
- [19] Rui Li, Zach Shahn, Jun Li, Mingyu Lu, Prithwish Chakraborty, Daby Sow, Mohamed Ghalwash, and Li-wei H Lehman. G-net : a deep learning approach to g-computation for counterfactual outcome prediction under dynamic treatment regimes. *arXiv preprint arXiv :2003.10551*, 2020.
- [20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco : Common objects in context. In *Computer Vision–ECCV 2014 : 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.
- [21] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.
- [22] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. *arXiv preprint arXiv :2204.07258*, 2022.

- [23] Nicola Messina, Giuseppe Amato, Fabio Carrara, Claudio Gennaro, and Fabrizio Falchi. Recurrent vision transformer for solving visual reasoning problems. In *International Conference on Image Analysis and Processing*, pages 50–61. Springer.
- [24] Judea Pearl and Dana Mackenzie. *The book of why : the new science of cause and effect*. Basic Books, 2018.
- [25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.
- [26] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, and Jost Tobias Springenberg. A generalist agent. *arXiv preprint arXiv :2205.06175*, 2022.
- [27] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union : A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666.
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.
- [29] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer.
- [30] K. Saketh. An intuitive explanation of self attention, 2020.
- [31] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond : A review of methods and applications. *Proceedings of the IEEE*, 109(3) :247–278, 2021.
- [32] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv :1907.01657*, 2019.
- [33] Jiajia Tang, Kang Li, Ming Hou, Xuanyu Jin, Wanzeng Kong, Yu Ding, and Qibin Zhao. Mmt : Multi-way multi-modal transformer for multimodal learning.

- [34] C. Thomas. U-nets with resnet encoders and cross connections, 2019.
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- [36] L-X Wang and Jerry M Mendel. Back-propagation fuzzy system as nonlinear dynamic system identifiers. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems*, pages 1409–1418. IEEE.
- [37] W. J. Wang. New similarity measures on fuzzy sets and on elements. *Fuzzy sets and systems*, 1997.
- [38] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, and Subhojit Som. Image as a foreign language : Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv :2208.10442*, 2022.
- [39] Xufei Wang and Jeongyoung Song. Iciou : Improved loss based on complete intersection over union for bounding box regression. *IEEE Access*, 9 :105686–105695, 2021.
- [40] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? *arXiv preprint arXiv :1905.13211*, 2019.
- [41] Jihong Yan and Zipeng Wang. Yolo v3+ vgg16-based automatic operations monitoring and analysis in a manufacturing workshop under industry 4.0. *Journal of Manufacturing Systems*, 63 :134–142, 2022.
- [42] L. A. Zadeh. Fuzzy sets. *Information and control*, 1965.
- [43] J. Zhang. Unet — line by line explanation, 2019.
- [44] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085.