

REPUBLIQUE DU CAMEROUN
Paix-Travail-Patrie

MINISTERE DE L'ENSEIGNEMENT
SUPERIEUR

UNIVERSITE DE DSCHANG

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES
&
INFORMATIQUE



REPUBLIC OF CAMEROON
Peace-Work-Fatherland

MINISTRY OF HIGHER EDUCATION

UNIVERSITY OF DSCHANG

FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS
&
COMPUTER SCIENCES

Détection de nids de poules sur le tronçon Dschang-Bafoussam

Par :

Nom et Prénom

DONGMO FEUDJIO Divin Virlence
KENMOGNE Ange Prisca

Matricule

CM-UDS-21SCI0878
CM-UDS-21SCI

Sous la direction de :

Nom et Prénom

Pr KENGNE TCHENDJI Vianney
YAWENI

Statut

Maître de Conférences
Chargé de TD

- 1 Analyse du papier : You only look once : Unified, real-time object detection
- 2 Collecte des données pour le projet

Analyse du papier : You Only Look Once (YOLO) I

Résumé et Contexte

Papier de Redmon et al. (2016) : Introduit YOLO pour détection en temps réel.

- **Approche** : Régression unique vs. pipelines complexes (R-CNN).
- **Points clés** :
 - Vitesse : 45 FPS (base), 155 FPS (Fast YOLO).
 - Précision : Double mAP temps réel, mais + erreurs localisation.
 - Généralisation : Meilleur sur art vs. DPM/R-CNN.

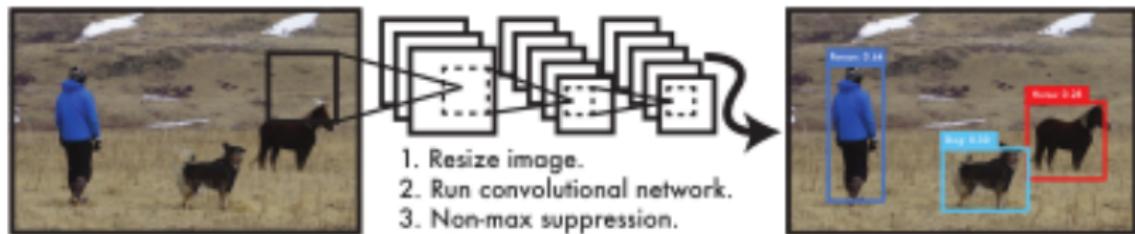


Figure – Système de détection de YOLO [1]

- Redimensionner l'image à 448x448
- Exécuter un réseau convolutif unique
- Seuiller les detections par confiance

Analyse du papier : You Only Look Once (YOLO)

Introduction (Section 1)

Problématique : Système humain rapide et précis vs. méthodes lentes (DPM, R-CNN) qui réutilisent des capteurs pour la détection.

- **YOLO** : Une passe pour prédictions objets/positions.
- **Avantages** :
 - Vitesse : 45-150 FPS, latence <25 ms.
 - Raisonnement global : $\frac{1}{2}$ faux positifs arrière-plan vs. Fast R-CNN.
 - Généralisation : Domaines inattendus (art).
- **Limites** : Localisation imprécise petits objets.

Analyse du papier : You Only Look Once (YOLO) I

Détection Unifiée (Section 2)

Architecture : Grille $S \times S$ et si le centre d'un objet tombe dans une cellule de grille, cette cellule de grille est responsable de la détection de cet objet.

- Chaque **cellule de grille** prédit B boîtes englobantes, des scores de confiance pour ces boîtes et C probabilités de classes conditionnelles.
- **Score** : $Pr(Class_i|Object) \times Pr(Object) \times IOU$.
- Chaque boîte englobante consiste en 5 prédictions : **x, y, w, h, et la confiance**
 - x,y : les coordonnées du centre de la boîte
 - w,h : la largeur et la hauteur de la boîte
 - la confiance : IOU entre la boîte prédite et toute boîte véritable terrain

Analyse du papier : You Only Look Once (YOLO) II

Détection Unifiée (Section 2)

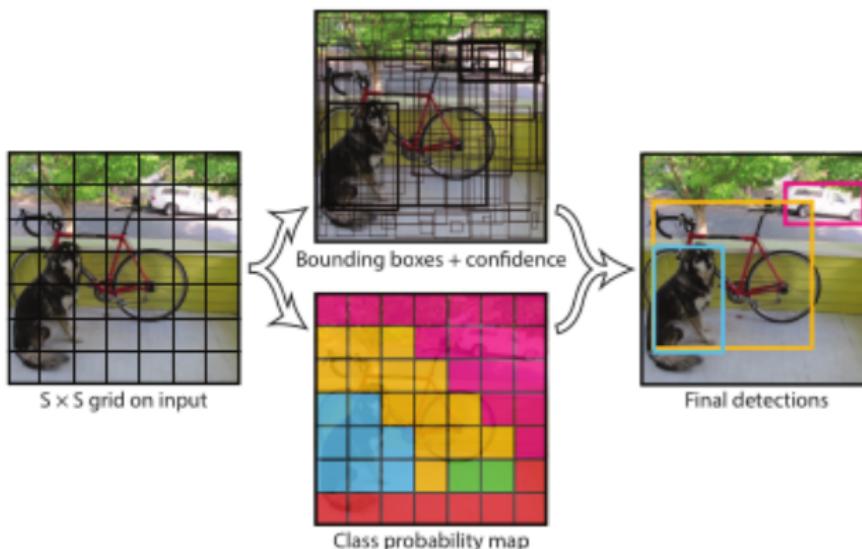


Figure – Modèle de détection de YOLO [1]

Analyse du papier : You Only Look Once (YOLO) III

Détection Unifiée (Section 2)

- x,y : les coordonnées du centre de la boîte
- w,h : la largeur et la hauteur de la boîte
- la confiance : IOU entre la boîte prédite et toute boîte vérité terrain
- **Réseau** : ce modèle est implémenté comme un réseau neuronal convolutif.

Analyse du papier : You Only Look Once (YOLO) IV

Détection Unifiée (Section 2)

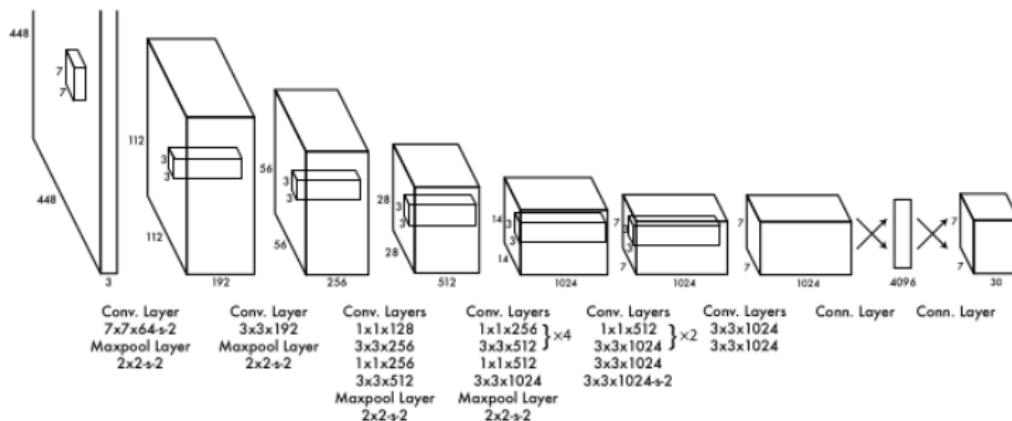


Figure – Modèle de détection de YOLO [1]

● Entrainement :

- Pré-entraînement des couches convolutives sur ImageNet (1000 classes).

Analyse du papier : You Only Look Once (YOLO) V

Détection Unifiée (Section 2)

- Conversion du modèle pour la détection en ajoutant 4 couches convolutives et 2 couches entièrement connectées.
- Augmentation de la résolution d'entrée de 224x224 à 448x448 pour des informations visuelles fines.
- Utilisation d'une fonction de perte somme des carrés pondérée ($\lambda_{coord} = 5$, $\lambda_{noobj} = 0.5$).
- Prédiction de la racine carrée de w et h pour mieux gérer les erreurs sur petites boîtes.
- Assignation d'un prédicteur responsable par objet basé sur l'IOU le plus élevé.
- **Inférence :**
 - Une seule évaluation du réseau pour prédire 98 boîtes et classes par image.
 - Utilisation de NMS pour éliminer les détections multiples (+2-3% mAP).

Analyse du papier : You Only Look Once (YOLO) VI

Détection Unifiée (Section 2)

- Pas de pipeline complexe, rapide pour le streaming vidéo.
- **Limites :**
 - Contraintes spatiales : seulement 2 boîtes et 1 classe par cellule, limite les objets proches.
 - Difficulté avec petits objets groupés (ex. : troupeaux d'oiseaux).
 - Généralisation limitée à ratios d'aspect ou configurations inhabituelles.
 - Caractéristiques grossières dues au downsampling.
 - Fonction de perte traite erreurs identiquement pour petites/grandes boîtes (source principale d'erreurs de localisation).

Analyse du papier : You Only Look Once (YOLO) I

Comparaison systèmes (Section 3)

YOLO vs. concurrents :

- **YOLO vs. DPM** : Architecture unifiée vs. approche de fenêtre glissante ; remplace extraction statique, classification et prédiction par un seul CNN optimisé pour la détection.
- **YOLO vs. R-CNN** : Contraintes spatiales sur propositions (98 vs. 2000) ; optimisation conjointe vs. pipeline disjoint lent (>40s/image).
- **YOLO vs. Fast/Faster R-CNN** : Plus rapide (45 FPS vs. 7 FPS) mais moins précis ; ne repose pas sur Selective Search.
- **YOLO vs. MultiBox** : Système complet vs. seulement prédiction de régions ; détection générale vs. pièce de pipeline.
- **YOLO vs. OverFeat** : Raisonnement global vs. local ; optimise pour détection vs. localisation.

Analyse du papier : You Only Look Once (YOLO) II

Comparaison systèmes (Section 3)

- **YOLO vs. MultiGrasp** : Détection multiple objets/classes vs. une seule région saisissable.
- **Tableau comparatif :**

Méthode	mAP	FPS
Fast YOLO	52.7	155
YOLO	63.4	45
Fast R-CNN	70.0	0.5

Analyse du papier : You Only Look Once (YOLO) I

Expériences (Section 4)

Résultats VOC 2007 :

- **4.1 Temps réel** : YOLO 63.4 mAP @45 FPS (2x DPM).
- **4.2 Erreurs** :
 - YOLO : 19% loc., 4.75% bg.
 - Fast R-CNN : 8.6% loc., 13.6% bg.
- **4.3 Combinaison** : +3.2% mAP (75.0%).
- **4.4 VOC 2012** : 57.9 mAP (faible petits objets).
- **4.5 Généralisation** : Art Picasso 59% AP (meilleur R-CNN).

Analyse du papier : You Only Look Once (YOLO) I

Sections 5-6 & Applicabilité

Section 5 (Détection en temps réel dans le monde réel) :

- YOLO est connecté à une webcam pour démontrer ses performances en temps réel, y compris la récupération des images et l'affichage des detections.
- Le système est interactif : traite les images individuellement mais fonctionne comme un suivi en détectant les objets en mouvement.
- Démo et code source disponibles sur le site du projet (open-source).

Section 6 (Conclusion) :

- YOLO est un modèle uniifié, simple à construire, entraîné directement sur images complètes.

Analyse du papier : You Only Look Once (YOLO) II

Sections 5-6 & Applicabilité

- Optimisé pour performances de détection ; rapide, précis et généralisable.
- Pousse l'état de l'art en détection temps réel ; idéal pour applications nécessitant robustesse.

- 1 Analyse du papier : You only look once : Unified, real-time object detection
- 2 Collecte des données pour le projet

Collecte des données I

Stratégie adoptée : Option 2 – Datasets publics + affinage local

Objectif : Obtenir un ensemble de données adapté aux routes camerounaises (tronçon Dschang-Bafoussam), avec au moins 500 images annotées au format YOLO.

- **Datasets publics (base) :**

- Dataset principal : **Potholes-Detection-YOLOv8** (Kaggle, par Angga Dwi Sunarto).
- Nombre d'images : 1581 (train) + 396 (validation).
- Classes : pothole (1 classe).
- Format : YOLO (dossiers train/images, train/labels ; valid/images, valid/labels ; fichier data.yaml prêt).
- Avantages : Images haute résolution, diverses conditions (éclairage jour/nuit, sec/humide), déjà annotées avec bounding boxes.
- Source : <https://www.kaggle.com/datasets/anggadwisunarto/potholes-detection-yolov8>

Collecte des données II

Stratégie adoptée : Option 2 – Datasets publics + affinage local

- Collecte locale complémentaire (affinage) :

- Objectif : Au moins 200 images spécifiques au Cameroun pour adapter le modèle aux caractéristiques locales (asphalte dégradé, poussière rouge, végétation tropicale, pluies intenses, ombres d'arbres).
- Méthode automatisée : Script Python utilisant **Google Custom Search JSON API** pour télécharger des images web pertinentes.
- Requêtes utilisées (queries) :
 - “routes Cameroun nids de poule”
 - “potholes Cameroon roads”
 - “Bafoussam bad road potholes”
 - “potholes Dschang road”
- Résultat : 200 images téléchargées.
- Script clé (extrait) :

Collecte des données III

Stratégie adoptée : Option 2 – Datasets publics + affinage local

- API : Google Custom Search (clé API + CX).
- Téléchargement progressif avec tqdm et gestion d'erreurs.
- Filtre : images larges (imgSize=large).
- **Annotation des images locales :**
 - Outil principal : **Roboflow** :
<https://app.roboflow.com/pothole-detection-wipyl/pothole-detection-dschang-bafous/1>
- **Diversité recherchée :**
 - Conditions météo : sec, pluvieux (flaques d'eau).
 - Éclairage : matin, midi, soir/nuit.
 - Types de routes : bitumées dégradées, secondaires, zones urbaines/périurbaines (Bafoussam, Dschang, axes ouest).

Références I

- [1] Joseph Redmon et al. "You only look once : Unified, real-time object detection". In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 779-788.