

Модуль реагування на пішоходів в системі автомобільного автопілоту

Автор : студент 4го курсу
групи КА-45
Харченко Д.О.

АКТУАЛЬНІСТЬ РОБОТИ

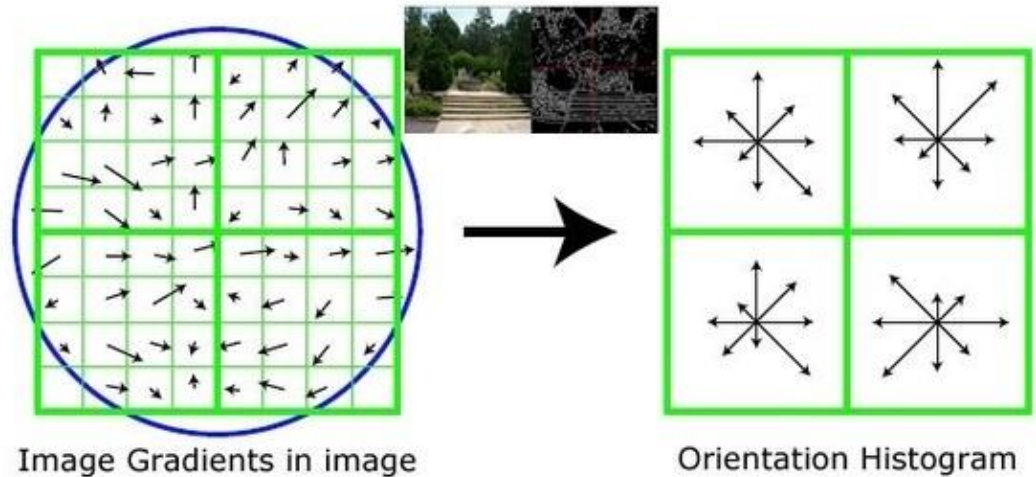
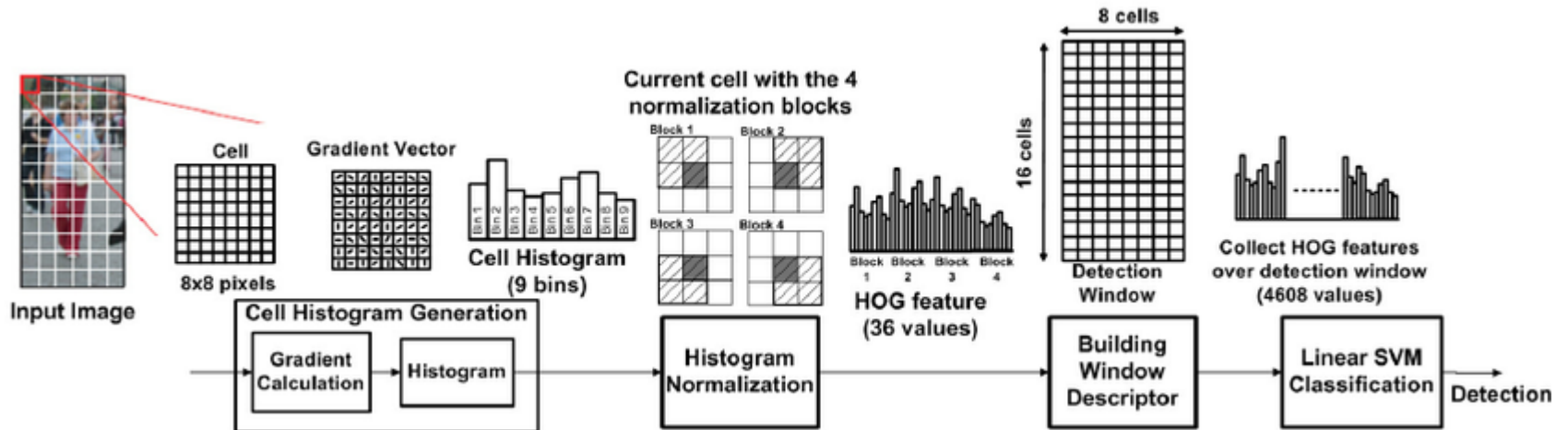
- ▶ Зменшення кількості та важкості ДТП за участі пішоходів
- ▶ Покращення якості роботи безпілотних автомобілів
- ▶ Системи оповіщення водія про небезпеку
- ▶ Системи автоматичного гальмування при надзвичайній ситуації

- ▶ **Об'єкт:** Системи комп'ютерного зору.
- ▶ **Предмет:** Моделі детектування об'єктів на зображенні. Класичні моделі, згорткові мережі.
- ▶ **Мета:** Аналіз методів та створення програми для детектування пішоходів на зображенні в реальному часі.

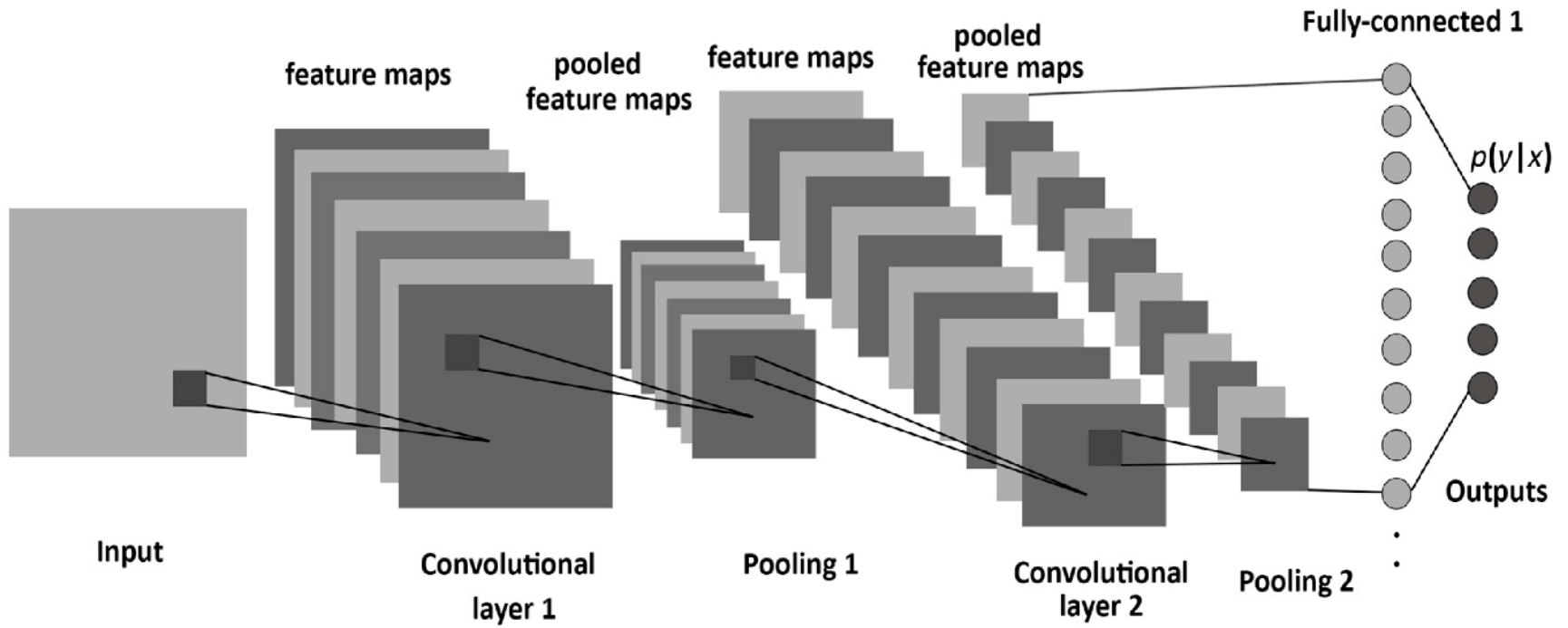
ПОСТАНОВКА ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

- ▶ Проаналізувати існуючі підходи та методи комп'ютерного зору, які доцільно використовувати при розпізнаванні пішоходів. Обрати метод для розробки ПП.
- ▶ Реалізувати ПП на основі обраного методу.
- ▶ Зробити експериментальні дослідження розробленої системи
- ▶ Сформулювати висновки щодо роботи алгоритму та можливості його покращення

КЛАСИЧНІ МОДЕЛІ

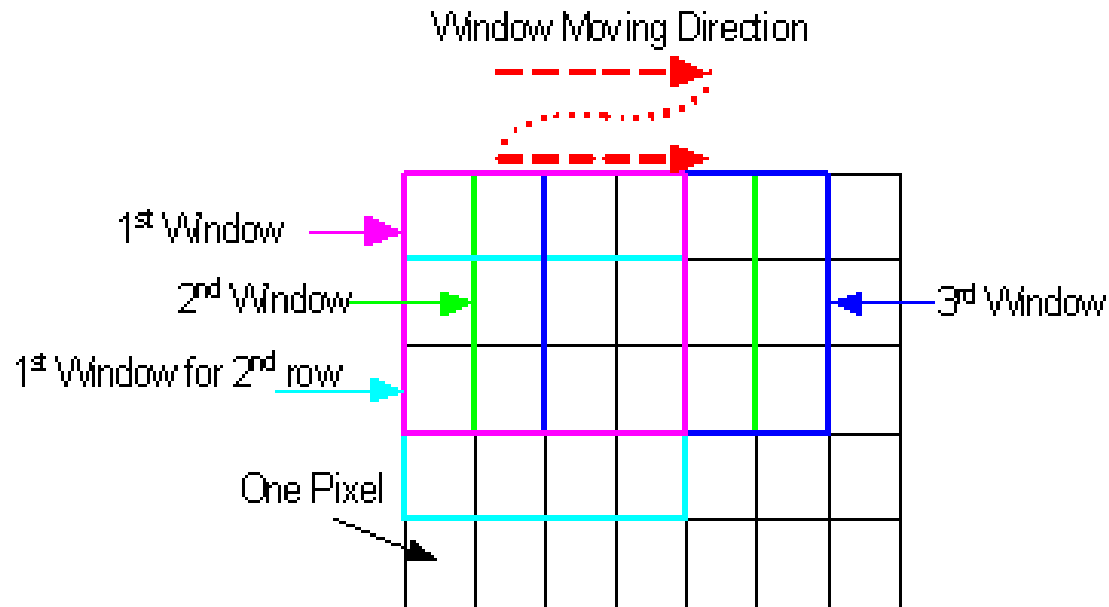


ЗГОРТКОВІ МЕРЕЖІ

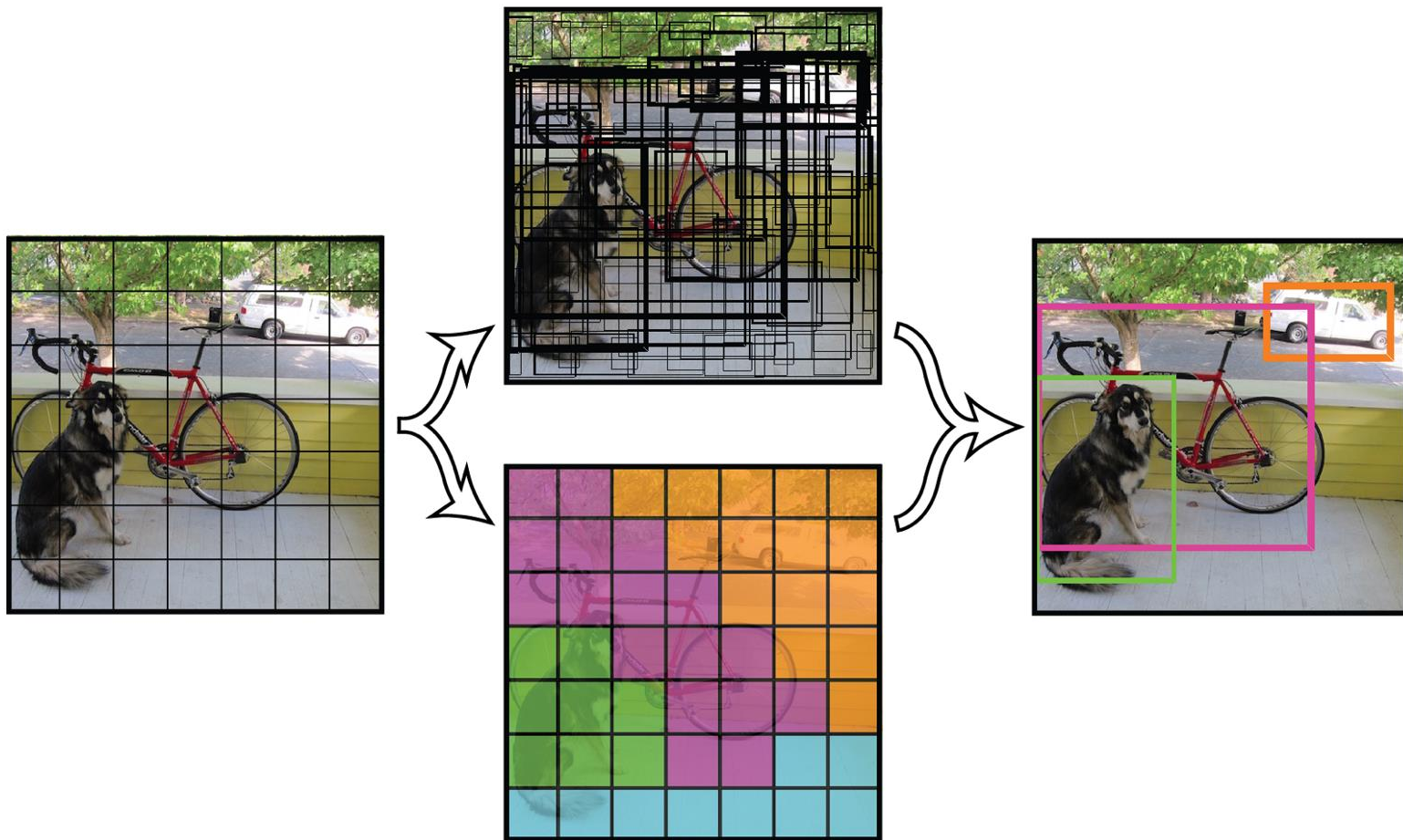


МЕТОД КОВЗКОГО ВІКНА

- ▶ В загальному вигляді суть методу полягає в багаторазовій класифікації частин зображення певного розміру з певним кроком.



МЕТОД ДЕТЕКЦІЇ ОБ'ЄКТІВ «ЗА ОДИН ПРОХІД» YOLO



LOSS FUNCTION

- ▶ Розмірність результату мережі = $S \times S \times B(5+C)$, де S – розмірність сітки, B – кількість можливих детекцій в одній комірці, C – кількість класів, що розпізнаються

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \sum_{c \in \text{classes}} (p_{ij}(c) - \hat{p}_{ij}(c))^2 \end{aligned}$$

Критерії оцінки

TP = True positive

TN = True negative

FP = False positive

FN = False negative

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Rank	Precision	Recall
Top 1	1.0	0.2
Top 2	1.0	0.4
Top 3	0.66	0.4
Top 4	0.75	0.6
Top 5	0.6	0.6
Top 6	0.66	0.8
Top 7	0.57	0.8
Top 8	0.5	0.8
Top 9	0.44	0.8
Top 10	0.5	1.0

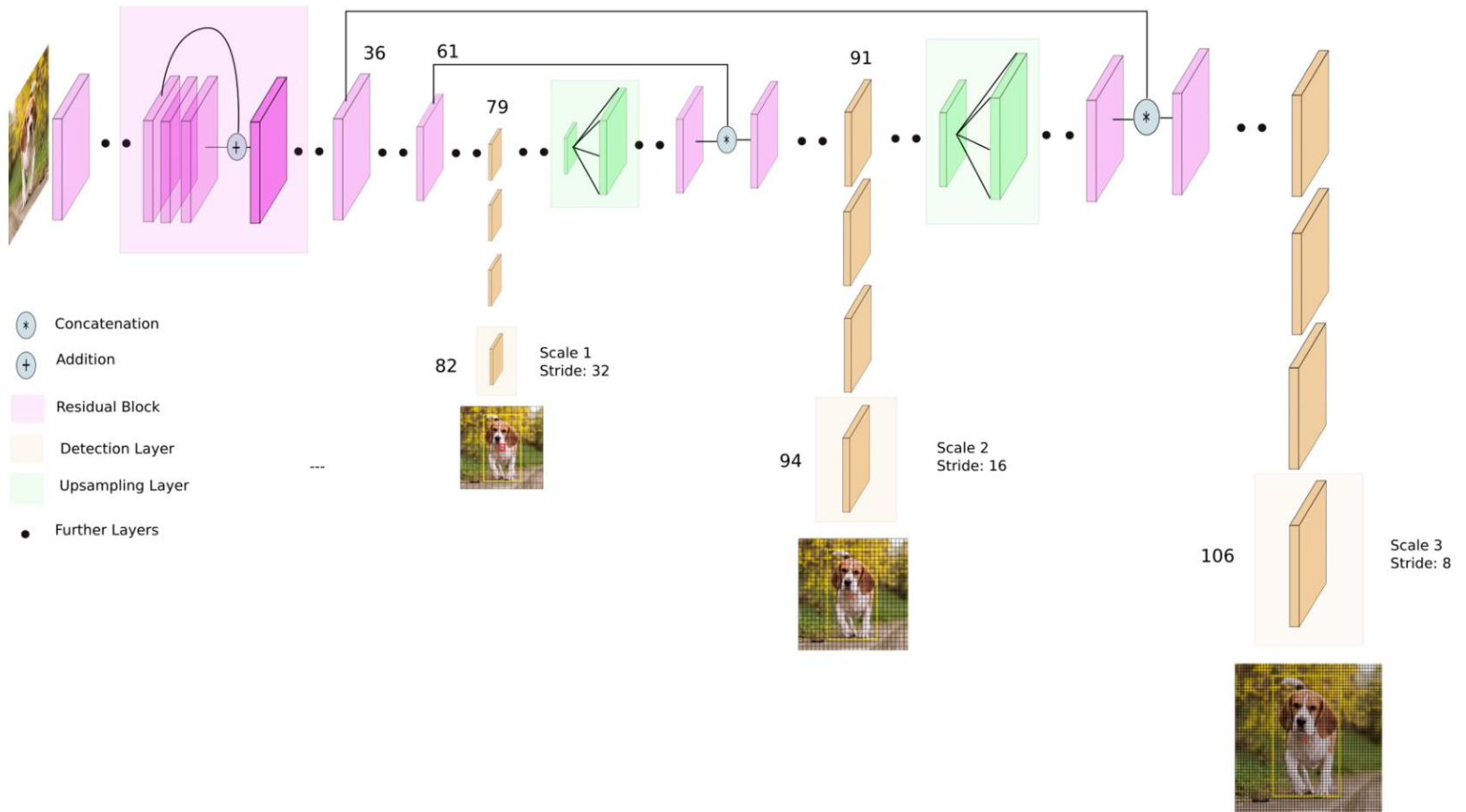
$$AP = \frac{1}{11} \sum_{r \in \{0.0, \dots, 1.0\}} AP_r = \frac{1}{11} \sum_{r \in \{0.0, \dots, 1.0\}} p_{interp}(r)$$

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

Існуючі методи

Детектор	Miss rate за Caltech	mAP за COCO IoU>0.5	mAP за VOC 2012	FPS
HOG+SVM	0.68	-	-	0.239
YOLO	-	-	63.5	45
YOLO2	-	44.0	81.3	67
R-CNN	-	-	72.0	0.5
Faster R-CNN	-	42.7	79.6	7

АРХИТЕКТУРА МЕРЕЖИ



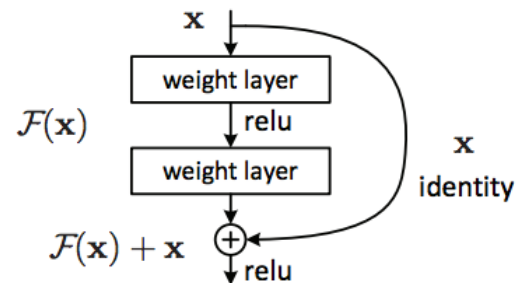
DarkNet

DarkNet19

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

DarkNet53

Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
Convolutional	32	1 × 1	128 × 128
Convolutional	64	3 × 3	
Residual			
Convolutional	128	3 × 3 / 2	64 × 64
Convolutional	64	1 × 1	64 × 64
Convolutional	128	3 × 3	
Residual			
Convolutional	256	3 × 3 / 2	32 × 32
Convolutional	128	1 × 1	32 × 32
Convolutional	256	3 × 3	
Residual			
Convolutional	512	3 × 3 / 2	16 × 16
Convolutional	256	1 × 1	16 × 16
Convolutional	512	3 × 3	
Residual			
Convolutional	1024	3 × 3 / 2	8 × 8
Convolutional	512	1 × 1	8 × 8
Convolutional	1024	3 × 3	
Residual			
Avgpool		Global	
Connected		1000	
Softmax			



Порівняння модифікації

Детектор	mAP за COCO IoU>0.5	FPS
YOLO2	44.0	2.5
Модифікований YOLO2	55.3	1.2



Архітектура ПП



РЕЗУЛЬТАТ РОБОТИ



ВИСНОВКИ

- ▶ Запропоновано модифікацію архітектури YOLO2, що покращує якість детекції, проте збільшує час обробки зображення.
- ▶ Розроблено програмний продукт на основі запропонованої модифікації, який показав високу точність на сьогодні та можливість подальшого покращення.
- ▶ Проведено аналіз результатів, який показав що досягнута достатня точність та повнота детекції для подальшої обробки отриманих результатів, наприклад, трекінг.

РЕКОМЕНДАЦІЇ ДЛЯ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- ▶ Реалізація детектору на GPU для покращення швидкодії
- ▶ Зменшення кількості класів розпізнавання для поліпшення швидкодії
- ▶ Доповнення функціоналом визначення відстані

ДЯКУЮ ЗА УВАГУ!