ZAŁĄCZNIK NR 2
do zarządzenia nr 54/2020 Rektora Politechniki Śląskiej
z dnia 13 marca 2020 r.

Pełnomocnik ds. Studenckiego Ruchu Naukowego
za pośrednictwem Centrum Obsługi Studiów

SPRAWOZDANIE
z realizacji projektu w ramach studenckiego koła naukowego

Przedkładamy sprawozdanie z projektu realizowanego w ramach studenckiego koła naukowego: **Linuks i Wolne Oprogramowanie.**

1. Temat projektu: Predykcja i klasyfikacja obiektów na zdjęciach, z wykorzystaniem metod sztucznej inteligencji.
2. Zespół projektowy

imię i nazwisko	kierunek studiów członka zespołu	poziom studiów	semestr
Maciej Król	Informatyka	SSI	7
Piotr Bosowski	Informatyka	SSI	7
Marcin Kaspryk	Informatyka	SSI	7

3. Lider zespołu (imię i nazwisko): Maciej Król
   E-mail: maciej.krol771@gmail.com
4. Opiekun naukowy (imię i nazwisko) dr hab. inż. Adam Domański, Prof. PŚ
   Jednostka organizacyjna: Katedra Systemów Rozproszonych i Urządzeń Informatyki
   E-mail: Adam.Domanski@polsl.pl
5. Przyjęte założenia: Badania realizowane w ramach tego projektu, są częścią większej pracy dążącej do stworzenia koncepcji autonomicznego pojazdu. W ramach wykonywanych prac chcemy stworzyć mechanizm detekcji popularnych obiektów ze zdjęć drogowych. Tworzony mechanizm pozwoli na wykrywanie osób, samochodów oraz rowerów, niezależnie od aktualnej pory dnia.
6. Osiągnięte cele:
   Stworzono i wytrenowano sieć neuronową służącą do detekcji i klasyfikacji obiektów na obrazach.
7. Zastosowane metody realizacji
   Do zrealizowania projektu sieci zastosowano bibliotekę “Darknet” [1] na licencji open-source.
8. Osiągnięte wyniki:
   Udało się osiągnąć dokładności wahające się od około 60% do około 90% dokładności klasyfikacji w zależności od klasy wykrywanego obiektu oraz algorytmu wstępnego przetwarzania zdjęcia.
9. Osiągnięte kamienie milowe:
   Analiza dostępnych rozwiązań. Przygotowanie danych wejściowych pochodzących ze zbioru FLIR [2]. Implementacja projektu w oparciu o architekturę YOLO (You Only Look Once) [11], udostępnianą w ramach biblioteki “Darknet”. Dostrojenie hiperparametrów modelu (ang. *fine-tuning*).

10. Formy upowszechniania wyników (publikacje, prezentacje, wystawy itp.) :

.....

11. Inne informacje o projekcie:

.....

Poniesiony wydatek	Data poniesienia wydatku

Podpisy członków zespołu:

1. .....
2. .....
3. .....

Opinia opiekuna naukowego:

.....

.....

.....

.....

.....

data i podpis opiekuna

Spis Treści

1. Wstęp teoretyczny

Uczenie maszynowe jest obszarem sztucznej inteligencji, który zajmuje się algorytmami automatycznie poprawiającymi zdolność komputerów do rozwiązywania problemów. Algorytmy te analizują istniejące dane i wyniki badań, by stworzyć mechanizm pozwalający na wykonanie analizy nowych danych wejściowych. Od lat sześćdziesiątych dwudziestego wieku, początków praktycznego zastosowania uczenia maszynowego, obserwowana jest coraz większa popularność tego obszaru. Na rozwój sztucznej inteligencji ogromny wpływ ma zwiększenie mocy obliczeniowej komputerów oraz internetu. Obecnie algorytmy uczenia maszynowego mogą być testowane i rozwijane nawet na najbardziej podstawowych zestawach komputerowych.

Spośród wielu możliwych zastosowań uczenia maszynowego, największą popularność zdobyło rozwiązywanie problemów związanych wizją komputerową i rozpoznawaniem wzorców [6]. Ma to związek z trudnością stworzenia odpowiednio dobrych algorytmów heurystycznych do klasyfikacji obiektów, ale również prostotą znalezienia przyzwoitego rozwiązania z wykorzystaniem technik sztucznej inteligencji. Z tych rozwiązań korzysta między innymi branża automotive. Uczenie maszynowe pozwoliło również na stworzenie “sztucznego nosa”, czy mechanizmów diagnostyki wielu chorób. Wyliczyć można by wiele innych przykładów zastosowań sztucznej inteligencji, uwzględniając obszar finansów, bezpieczeństwa, transportu i wiele innych.

2. Cel projektu

Projekt „Predykcja i klasyfikacja obiektów na zdjęciach, z wykorzystaniem metod sztucznej inteligencji” miał na celu stworzenie rozwiązania, które zapewniłoby detekcję i klasyfikację obiektów na obrazach generowanych przez kamerę termowizyjną. Koło naukowe SKNLiWO jest w posiadaniu zestawu zdjęć, który potencjalnie mógłby posłużyć za zbiór treningowy dla sieci neuronowych [2].

W ramach projektu oczekiwane jest rozwiązanie, które pozwoli na odtworzenie istniejących rozwiązań w warunkach laboratoryjnych oraz będzie stanowiło dodatkowy wkład w badania nad skutecznością zastosowań zdjęć termowizyjnych w branży automotive. Ponadto zakłada się sprawdzenie produktu finalnego, w postaci sieci neuronowej w warunkach rzeczywistych. Przeprowadzone zostaną również badania mające na celu optymalizację rozwiązania, poprzez porównanie różnych wariacji możliwych rozwiązań. Jako główny cel postawiono sobie realizację modelu pozwalającego na jak najbardziej efektywną predykcję wybranych obiektów. Wybrano w tym celu predykcję samochodu, roweru oraz człowieka.

3. Środowisko eksperymentalne

Stacja robocza, na której przeprowadzone zostały obliczenia, wyposażona jest w procesor AMD Ryzen Threadripper 1920X, kartę graficzną Nvidia RTX 2070, 96 GB pamięci RAM oraz system operacyjny Windows 10. Zostały na niej zainstalowane wszystkie pakiety konieczne do skompilowania i uruchomienia frameworka Darknet, w szczególności pakiet CUDA 11.0 (biblioteka zrównoleglania obliczeń na karcie graficznej) oraz OpenCV 4.4.0 (biblioteka ułatwiająca przetwarzanie obrazów stosowana w dziedzinie wizji komputerowej) [7].

4. Biblioteka Darknet z systemem YOLO

System YOLO (*ang. you only look once*) [11], zaimplementowany w ramach biblioteki Darknet, pozwala na detekcję obiektów w czasie rzeczywistym z wydajnością przewyższającą klasyczne metody wykorzystujące detekcję dwuetapową. Na karcie graficznej Nvidia Titan X (architektura Pascal) przetwarza obrazy z szybkością 30 kl./s i posiada średnią precyzję predykcji (*ang. mean average precision*, mAP) na poziomie 57.9% na referencyjnym zbiorze COCO [9]. Opisywany system wykorzystuje pojedynczą sieć neuronową dla całego obrazu, która dzieli obraz na regiony i przewiduje wystąpienie w tych regionach obiektów [12], minimalizując w ten sposób czas potrzebny na przetworzenie jednej klatki, co bezpośrednio przekłada się na wydajność całego rozwiązania.

5. Przygotowanie danych

Kolejny etap polegał na wstępnym przygotowaniu danych treningowych, walidacyjnych i testowych. Wybrana przez nas biblioteka wymaga utworzenia pliku zawierającego w kolejnych wierszach ścieżki do wszystkich zdjęć należących do danego podzbioru (treningowego/walidacyjnego/testowego). Przykład kodu przygotowującego zbiór treningowy zaprezentowano na grafice 1.

```
with open("train.txt", 'w') as file:
    folder_path = 'FLIR_ADAS_1_3/train/Annotated_thermal_8_bit'
    current = os.getcwd()
    images = [image for image
              in os.listdir(folder_path)
              if image.endswith('.jpeg')]
    for image in images:
        file.write(os.path.join(os.getcwd(), folder_path, image))
        file.write('\n')
```

Rysunek 1. Kod przygotowujący zbiór danych treningowych dla biblioteki Darknet

```
for subset in ['train', 'val', 'video']:
    images_path = os.path.join('FLIR_ADAS_1_3', subset,
                               'Annotated_thermal_8_bit')
    annotations_path = os.path.join('FLIR_ADAS_1_3', subset,
                                    'thermal_annotations.json')
    with open(annotations_path, 'r') as json_file:
        border_dict = json.load(json_file)
    for image in border_dict['images']:
        current_annotations = [annot for annot in border_dict['annotations']
                               if annot['image_id'] == image['id']]
        image_name = os.path.basename(image['file_name'])
        base, ext = os.path.splitext(image_name)
        with open(os.path.join(images_path, base + ".txt"), 'w') as file:
            for annot in current_annotations:
                category = category_map[annot['category_id']]
                x = (annot['bbox'][0] + annot['bbox'][2] / 2) / image['width']
                y = (annot['bbox'][1] + annot['bbox'][3] / 2) / image['height']
                width = annot['bbox'][2] / image['width']
                height = annot['bbox'][3] / image['height']
                darknet_format = f"{category} {x} {y} {width} {height}\n"
                file.write(darknet_format)
```

Rysunek 2. Fragment programu konwertujący informacje o pozycji obiektów na zdjęciach ze zbioru FLIR do standardu biblioteki Darknet

```
<object-class> <x> <y> <width> <height>
```

Rysunek 3. Format zapisu informacji o pozycji obiektów na zdjęciu używany przez wybraną bibliotekę. Każde zdjęcie ma odpowiadający plik, w którym w kolejnych wierszach umieszcza się informację o kolejnych obiektach widocznych na danym zdjęciu. X, y, width *i* height *opisują pozycję i rozmiar obiektu.*

Ponadto dla każdego zdjęcia należy także wygenerować plik zawierający opis pozycji i kategorii obiektów znajdujących się na zdjęciu w formacie zdefiniowanym w dokumentacji biblioteki (rysunek 2). Ze względu na brak jednego standardu zapisu powyższych metadanych, konieczne jest każdorazowe przygotowanie kodu przekształcającego format użyty przez twórców zbioru danych do formatu akceptowanego przez daną bibliotekę. Kod realizujący konwersję ze zbioru FLIR do standardu frameworku Darknet pokazano na grafice 3.

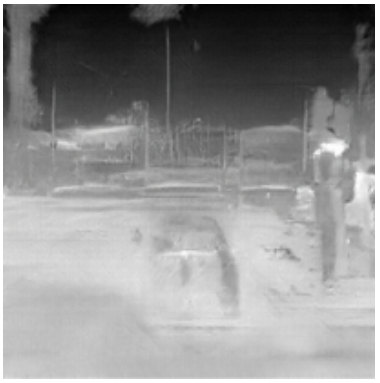
Zazwyczaj przed trenowaniem modelu konieczna jest wcześniejsza obróbka graficzna zdjęć [4]. Możliwym byłoby również zastosowanie technik segmentacji zdjęć przed wykorzystaniem ich w sieci neuronowej [5]. Obrazy ze zbioru nie wymagały znacznych zmian w tym zakresie, gdyż były dobrze przygotowane do wykorzystania w procesie uczenia sieci neuronowych.

6. Osiągi wytrenowanego modelu

Model został wytrenowany z wykorzystaniem 17 tysięcy różnych zdjęć, pozwalający na detekcję w czasie rzeczywistym trzech różnych obiektów z następującymi dokładnościami. Dla detekcji człowieka dokładność predykcji wyniosła 81.32%. W tym próbka stanowiąca poprawną predykcję wyniosła 4438 obiektów. Dla roweru dokładność predykcji wyniosła 60.57% w tym próbka stanowiąca poprawną predykcję wyniosła 269 obiektów. Dla ostatniego obiektu samochodu dokładność predykcji wyniosła najwięcej spośród wszystkich obiektów, a mianowicie 89.16% w tym próbka stanowiąca poprawną predykcję wyniosła 4742 obiekty.

7. Analiza eksperymentalna jakości detekcji

Niestety, z powodu braku dostępu do kamery termowizyjnej, która pozwoliłby na szczegółową analizę jakościową prezentowanego rozwiązania, także w czasie rzeczywistym, podjęto rozważania nad innymi możliwymi rozwiązaniami symulującymi obraz rejestrowany przez kamerę termowizyjną. Odnaleziono istniejące algorytmy przetwarzające zdjęcia z formatu RGB na termowizyjne [11], które mogłyby, po zaimplementowaniu w projekcie, zasymulować działanie brakującej kamery. Niestety takie podejście nie dawało większych szans na powodzenie, gdyż istniejące sieci neuronowe nie radzą sobie wystarczająco dobrze z konwertowaniem zdjęć, by te nadawały się do testowania sieci klasyfikującej wykonanej w ramach tego projektu [3]. Pomimo świadomości znanych rozwiązań, które zakończyły się niesatysfakcjonującymi wynikami postanowiono podjąć próbę zrealizowania tego zadania na zakupionym w ramach dofinansowania komputerze. W tym celu wykorzystano sieć *pix2pix* [8]. Uczenie, zgodnie z przewidywaniami wynikającymi z analizowanych źródeł [3] okazało się bezskuteczne, a wyniki nie mogły zostać wykorzystane. Przykładowy wynik działania sieci pokazuje rysunek 4.

Oryginalny obraz RGB	Wynik działania sieci *pix2pix*	Docelowy obraz termowizyjny
		

Rysunek 4. Przykładowy obraz otrzymany w wyniku transformacji przez model pix2pix.

Jak się okazało wydłużanie czasu uczenia o kolejne setki epok nie przynosiło efektów, a kolejne godziny, przez które karta graficzna usiłowała osiągnąć pożądany rezultat nie poprawiały wyników. Sumaryczny czas poświęcony na desperackie próby liczyć można w dziesiątkach godzin. Zważając na powyższe zarzucono to podejście,

tym samym skłaniając się ku zaleceniu by jeśli to możliwe, w takich wypadkach stosować po prostu heurystyczny algorytm skali szarości.

8. Podsumowanie

W ramach projektu przeanalizowano literaturę pod kątem istniejących rozwiązań. Przygotowano stację roboczą i zainstalowano komplet oprogramowania potrzebnego do skompilowania i uruchomienia systemu YOLO zaimplementowanego w ramach biblioteki Darknet. Przygotowano program przygotowujący dane wejściowe z biblioteki FLIR do przetworzenia przez bibliotekę (kod źródłowy dostępny pod adresem github.com/PiotrBosowski/thermobits). Dokonano dostrojenia hiperparametrów modelu pod kątem maksymalizacji metryki mAP na zbiorze walidacyjnym. Zbadano możliwość wykorzystania istniejących sieci neuronowych do przekształcania zdjęć RGB w zdjęcia termowizyjne. Finalnie osiągnięto wysoką dokładność detekcji ludzi i samochodów (odpowiednio 81.32% i 89.16%) oraz zadowalającą dokładność detekcji rowerów (60.57%). Analiza jakościowa błędnych detekcji tej kategorii pokazała, że mimo niewykrycia samego roweru, jadący na nim człowiek był w dużej mierze wykrywany i klasyfikowany poprawnie.

Bibliografia

[1]	Darknet framework https://pjreddie.com/darknet/
[2]	FREE FLIR Thermal Dataset for Algorithm Training https://www.flir.com/oem/adas/adas-dataset-form/#:~:text=Why%20Use%20FLIR%20Thermal%20Sensing%20for%20ADAS%3F&text=The%20FLIR%20thermal%20sensors%20can,LiDAR%2C%20radar%20and%20visible%20cameras. [data dostępu: 20.12.2020]
[3]	Hannes Liik Thermal Image Generation from RGB https://medium.com/@hannesliik/thermal-image-generation-from-rgb-b152efa66cc2 [data dostępu: 20.12.20202]

[4]	Hongjun Lu Sam Yuan Sung Ying Lu. *On Preprocessing Data for Effective Classification* (1996)
[5]	Hussain Dar, Nasir. (2020). Image segmentation Techniques and its application.
[6]	Islam Hasabo. *Image Classification using Machine Learning and Deep Learning* https://medium.com/swlh/image-classification-using-machine-learning-and-deep-learning-2b18bfe4693f [data dostępu: 19.12.2020]
[7]	OpenCV documentation. https://docs.opencv.org/master/ [data dostępu: 19.12.2020]
[8]	Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros Image-to-Image Translation with Conditional Adversarial Networks (2018).
[9]	Tsung-Yi Lin et al., *COCO - Common Objects in Context dataset*, 2015
[10]	Vladimir V. Kniaz , Vladimir A. Knyaz , Jiˇr´ı Hlad˚uvka , Walter G. Kropatsch, and Vladimir Mizginov ThermalGAN: Multimodal Color-to-Thermal Image Translation for Person Re-Identification in Multispectral Dataset https://openaccess.thecvf.com/content_ECCVW_2018/papers/11134/Kniaz_ThermalGAN_Multimodal_Color-to-Thermal_Image_Translation_for_Person_Re-Identification_in_Multispectral_ECCVW_2018_paper.pdf
[11]	YOLO https://pjreddie.com/darknet/yolo/