



**Politechnika
Śląska**

Dokumentacja projektowa

MLA 2021/2022

**Aplikacje uczenia maszynowego
w systemach interakcji człowiek-maszyna**

Kierunek: Informatyka

Członkowie zespołu:

Dawid Bitner

Daniel Broczkowski

Damian Kwaśniok

Gliwice, 2021/2022

Spis treści

1	Wprowadzenie	2
1.1	Cel projektu	2
1.2	Zespół projektowy	2
2	Założenia projektowe	3
3	Implementacja	4
4	Badania	5
5	Podsumowanie i wnioski	8
6	Spis literatury	9

1 Wprowadzenie

1.1 Cel projektu

Celem projektu było wykonanie systemu opierającego się o interakcję człowiek-komputer. Zespół projektowy postanowił zaimplementować aplikację rozpoznającą, czy osoba stojąca przed kamerą ma założoną масечkę na twarzy. Projekt nosi nazwę „face mask detector”. Głównym celem projektu było stworzenie użytecznej aplikacji mogącej mieć zastosowanie w obszarze przestrzeni publicznej (uczelnie, urzędy, inne placówki publiczne) jak i mogącej być wykorzystywanej przez podmioty prywatne - projekt miał za zadanie pokazać w jaki sposób interakcja człowiek-komputer może być wykorzystywana w codziennym życiu.

1.2 Zespół projektowy

Dawid Bitner :: Deweloper :: Opracowanie obszaru ML i HCI, implementacja projektu końcowego.

Daniel Broczkowski :: Deweloper :: Opracowanie obszaru ML i HCI, implementacja projektu końcowego.

Damian Kwaśniok :: Deweloper :: Opracowanie obszaru ML i HCI, implementacja projektu końcowego.

2 Założenia projektowe

Do głównych zależeń projektowych określonych przez zespół implementujący rozwiązanie można zaliczyć m.in.:

- uniwersalność
- skalowalność
- możliwość ulepszeń
- wykorzystanie *Machine Learningu*
- oparcie o interakcję człowiek-komputer

Zdaniem zespołu odpowiedzialnego za projekt wszystkie założenia zostały spełnione. System detekcji maski na twarzy można uruchamiać na urządzeniach z różnymi systemami operacyjnymi np.; *Microsoft Windows, Linux*. Projekt może być uruchomiony na każdym środowisku, na którym został zainstalowany interpreter *Python 3* wraz z bibliotekami.

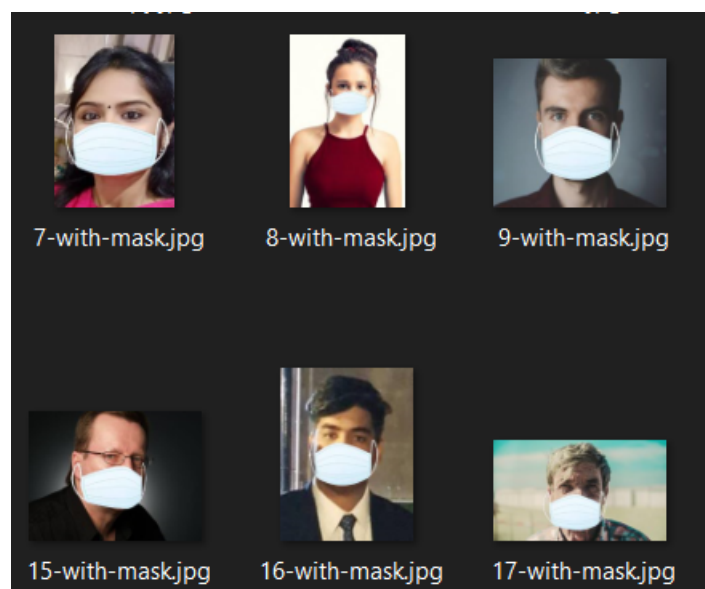
Program jest skalowalny oraz podatny na ulepszenia, może on być ciągle udoskonalany poprzez dodanie dodatkowych opcji w przyszłości np.: oprócz rozpoznawania maseczki na twarzy dodanie rozpoznawanie osoby przed kamerą. Szczególnie przydatne jeżeli oprócz weryfikacji noszenia maski potrzeba również weryfikować osobę przed kamerą, np.: czy jest nauczycielem akademickim, pracownikiem itp.

Program został przeszkolony w celu detekcji maski na twarzy za pomocą uczenia maszynowego. Zachodzi również interakcja człowiek-komputer, czy też człowiek-maszyna. Osoba podchodzi do kamery, oprogramowanie weryfikuje czy ma założoną maseczkę, następnie na podstawie wyniku działania algorytmu można podjąć decyzję co do wyniku.

3 Implementacja

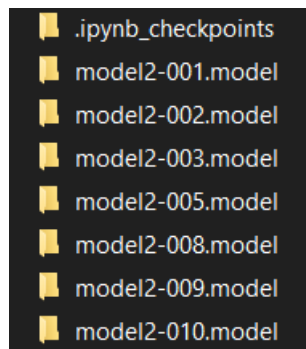
Projekt został zaimplementowany w oparciu o język programowania *Python*. wykorzystane zostały składowe takich bibliotek jak *TensorFlow*, *Keras*, *Numpy*.

W pierwszej kolejności zostały wybrany zbiór danych do przetrenowania. Trzeba podkreślić, że wybrany zbiór nie jest idealny, zostały w nim użyte tylko chirurgiczne maseczki o jasnej kolorystyce, oraz w niektórych przypadkach zostały one doklejone do zdjęcia za pomocą obróbki graficznej. Z tego też powodu, co zostanie przedstawione w kolejnym rozdziale system nie do końca poprawnie rozpoznaje sytuację w której osoba przed kamerą ma założoną ciemną maseczkę. Zdjęcia do treningu i testów zostały podzielone na dwie grupy: z maseczką i bez maseczki.



Rysunek 1: Część zbioru danych do treningu - z maseczką

Następnie został stworzony skrypt z siecią neuronową - 8 warstwową, która trenuje i testuje program. W ostatniej kolejności został napisany skrypt

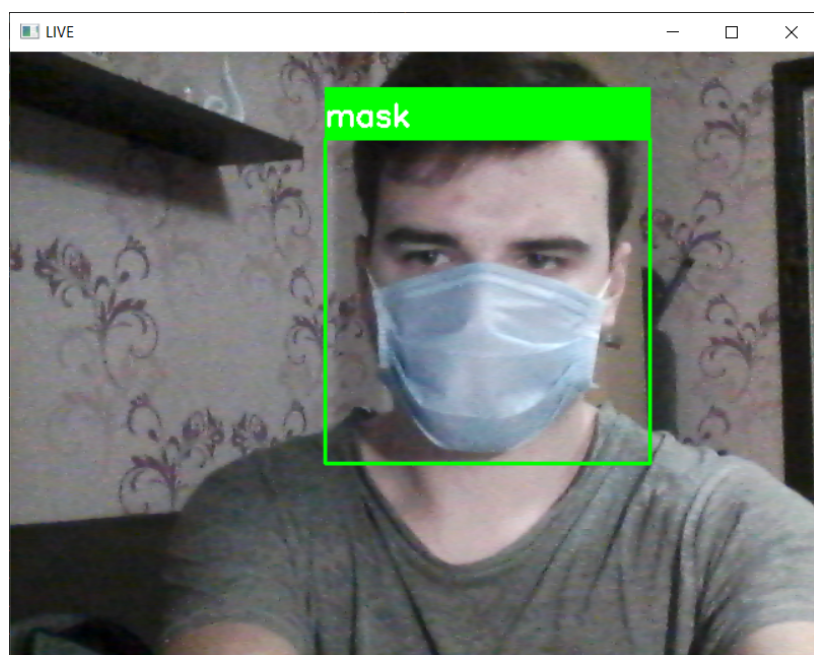


Rysunek 2: Modele wynikowe

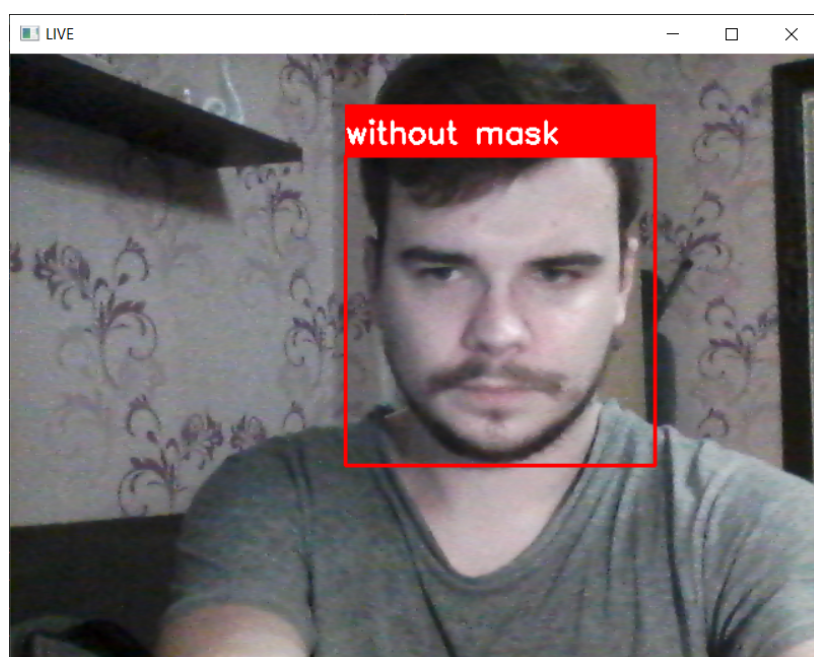
integrujący kamerę w urządzeniu z systemem wykrywania twarzy. Wyniki działania widoczne są w kolejnym rozdziale.

4 Badania

Zespół projektowy podczas wykonywania projektu sugerował się najnowszymi trendami w dziedzinie *Machine Learningu*, stworzona sieć działa poprawnie, nie obciąża mocno systemu operacyjnego, oraz na wybranym zbiorze danych działa w naszej opinii dosyć szybko. W teorii system powinien wykrywać przede wszystkim głowę osoby za pomocą klasyfikatora twarzy *haar-cascade_frontalface_default* - który w pliku *.xml* zawiera zbiór koordynatów opracowany przez firmę *Intel* na licencji *Open Source* - o czym informuje w komentarzu w pliku. Dodatkowo system sprawdza to, czy badany osobnik ma założoną maseczkę, czy też jej nie ma. W praktyce zadanie to nie zawsze jest spełniane w stu procentach, co przedstawione jest na poniższych przykładach.



Rysunek 3: Prawidłowo wykryta twarz z maseczką.



Rysunek 4: Prawidłowo wykryta twarz bez maseczki

Pierwsze dwa przypadki zostały prawidłowo rozpoznane. Jest to w głównej mierze zasługa dwóch czynników. Pierwszy z nich to dobre oświetlenie - badana twarz jest jasna, a jej kontury kontrastują z tłem. Drugim czynnikiem jest sam zbiór treningowy, który nie jest takiej jakości, jakiej zespół projektowy by sobie życzył. Nierzadko maseczki zostały doklejone do zdjęć w programach graficznych, przez co brakuje normalizacji sceny zdjęcia i maseczka wyraźnie kontrastuje na tle reszty głowy (warto zauważyć, iż w omawianym przypadku użyto maseczki o błękitnym kolorze).

5 Podsumowanie i wnioski

- *Podsumowanie*

System cechuje wysoką skutecznością. Biorąc pod uwagę zbiory testowe i treningowe wynosi ona ok. 92% biorąc pod uwagę wymóg testu w jasnej maseczce. Naszym zdaniem - jak na zbiór danych którym został przetrenowany, działa bardzo dobrze. System również poprawnie wykrywa istnienie twarzy.

- *Wnioski*

System jest prosty, łatwy do uruchomienia na różnych środowiskach, naszym zdaniem do tego powinno dążyć nowoczesne oprogramowanie. System niepoprawnie wykrywa ciemne maseczki, kolejna iteracja szkolenia systemu powinna przebiegać na nieco bardziej zróżnicowanych obrazach testowych (różnokolorowe maski). Z pewnością jest to najważniejszy obszar do przepracowania w tej implementacji.

6 Spis literatury

- [1] D. Bitner D. Broczkowski M. Kowol M.Krupa. „Image classification and convolutional neural networks”. W: *Silesian University of Technology: Faculty of Applied Mathematics* (2019). URL: https://github.com/dawbit/MeowFinder/blob/master/article/Artificial_Intelligence_Article.pdf.
- [2] Huilin Xiong M.N.S.Swamy M.O.Ahmad. „Two-dimensional FLD for face recognition”. W: *Elsevier* (2005). URL: <https://www.sciencedirect.com/science/article/abs/pii/S0031320305000130>.