Przetwarzanie strumienia wideo dla systemu elektronicznego zdrowia

*Rozpoznawanie emocji przy użyciu konwolucyjnej sieci neuronowej

Marta Budnik
wydział Elektroniki
Politechnika Wrocławska
Wrocław, Polska
235974@student.pwr.edu.pl

Maciej Dados
wydział Elektroniki
Politechnika Wrocławska
Wrocław, Polska
235320@student.pwr.edu.pl

Karolina Zdon
wydział Elektroniki
Politechnika Wrocławska
Wrocław, Polska
235107@student.pwr.edu.pl

Natalia Brychcy
wydział Elektroniki
Politechnika Wrocławska
Wrocław, Polska
235378@student.pwr.edu.pl

Adam Stanisławski
wydział Elektroniki
Politechnika Wrocławska
Wrocław, Polska
235540@student.pwr.edu.pl

Streszczenie—Korzystanie z możliwości jakie daje detekcja twarzy stało się w ostatnich latach bardzo popularne. W tym projekcie postanowiliśmy skupić się na analizie części informacji, które można otrzymać z ludzkiej twarzy - emocje oraz puls. Projekt ten opiera się na ostatnich badaniach w celu sklasyfikowania obrazów ludzkich twarzy w siedmiu emocjach przy użyciu konwolucyjnych sieci neuronowych (CNN). Model wyszkolony na zbiorze danych FER2013 osiąga dokładność równą 66%.

Index Terms—uczenie maszynowe, sieci splotowe, sieci konwolucyjne, CNN, FER2013, rozpoznanie wyrazu twarzy, emocje„ klasyfikatory kaskadowe Haara, detekcja twarzy, rozpoznanie w czasie rzeczywistym

I. Wstęp

Projekt ma na celu przetworzenie strumienia wideo, które umożliwi bezdotykowe monitorowanie stanu zdrowia osób. Informacje, które zostaną w tym celu przetworzone to puls oraz ludzkie emocje, które pozwolą ustalić stan zdrowia od strony kardiologicznej oraz psychicznej. Monitoring wizyjny będzie częścią systemu, który po przetworzeniu danych zwróci informację zwrotną badanej osobie. W tym celu posłuży specjalnie stworzona aplikacja webowa. W artykule skupimy się jednak na części związanej ze sztuczną inteligencją. W przypadku pulsu korelacja z metodami sztucznej inteligencji występuje tylko w momencie detekcji twarzy, zatem ta praca naukowa będzie poruszać temat związany z emocjami. Puls jest mierzony poprzez badanie zmian jasności fragmentów naszego ciała (czoła), które wynikają z przepływu krwi. Ciąg klatek poddawany jest transformacie Fouriera, która pozwala wyciągnąć częstotliwość tych zmian, czyli puls. Zgodnie z badaniami prowadzonymi przez zespół z Uniwersytetu w Kalifornii, dokładność takich pomiarów waha się między ± 5, a 15% w zależności od tego, przy jakim oświetleniu wykonywany jest pomiar oraz na ile osoba badana pozostaje nieruchomo.

A. Przegląd literatury

Zdolność rozpoznawania wyrazu twarzy jest kluczem do niewerbalnej komunikacji między ludźmi. Percepcja i interpretacja mimiki twarzy są szeroko badane, dlatego podejmowane są skuteczne próby uzyskania wniosków poprzez uczenie maszynowe. Automatyczne rozpoznawanie wyrazu twarzy pozostaje trudnym i interesującym problemem w komputerowej wizji. Rozpoznawanie mimiki twarzy jest trudnym problemem dla technik uczenia maszynowego, ponieważ ludzie mogą się znacznie różnić w sposobie, w jaki wyrażają swoją ekspresję. Głębokie uczenie się jest nowym obszarem badań w ramach metody uczenia maszynowego, która może klasyfikować obrazy ludzkich twarzy do kategorii emocji przy użyciu sieci neuronowych. [2] Podejścia metodyczne do tworzenia modelu wykrywania emocji u ludzi są różne. Metody statystyczne zwykle obejmują stosowanie różnych nadzorowanych algorytmów uczenia maszynowego, w których duży zestaw danych z komentarzami jest wprowadzany do algorytmów, aby system mógł się nauczyć i przewidzieć odpowiednie typy emocji. [3]

Podejście to zwykle obejmuje dwa zestawy danych: zestaw treningowy i zestaw testowy, ten pierwszy jest wykorzystywany do uczenia atrybutów danych, z kolei drugi służy do sprawdzania poprawności działania algorytmu uczenia maszynowego. Wyuczone maszynowo algorytmy zapewniają zazwyczaj bardziej rozsądną dokładność klasyfikacji w porównaniu z innymi podejściami, ale jednym z wyzwań w osiąganiu dobrych wyników w procesie kla-

syfikacji jest potrzeba posiadania wystarczająco dużego zestawu treningowego. [4]

Dobrze znane algorytmy głębokiego uczenia obejmują różne architektury sztucznej sieci neuronowej, takie jak konwolucyjna sieć neuronowa (CNN), czy sieci rekurencyjne (RNN). Ta pierwsza jest szeroko stosowana w najnowszych badaniach przezwyciężenia trudności w klasyfikacji wyrazu twarzy. [7] Praca ta będzie się koncentrować na ewaluacji wśród różnych grup etnicznych, gdyż jest to bardziej wymagający scenariusz rozpoznawania wyrazu twarzy.

Konwolucyjne sieci neuronowe mają potencjał, aby przezwyciężyć te wyzwania. Byoung Chul Ko w swojej pracy poświęconej rozpoznawaniu emocji porównywał metodę CNN wraz z LSTM. Operując na zbiorze danych o nazwie FER2013, otrzymał wyniki 72.65% dla CNN oraz 63.2% dla LTSM. [8] Prace dostępne w internecie na ten temat prezentują zazwyczaj wyniki w zależności od wieku badanego, oświetlenia otoczenia, pozycji twarzy, czy intensywności ekspresji, co oddaje realistyczne warunki. Materiały na ten temat różnią się znacznie pod względem architektury sieci neuronowych, przetwarzania wstępnego, podziału na zbiory szkoleniowe i testowe, a także czynnikami, które mają wpływ na wydajność. [9]

B. Teoretyczne zakreślenie tematu

Tematem tej pracy jest zbudowanie efektywnego modelu sieci neuronowej, który będzie w stanie wykrywać emocje z twarzy człowieka w czasie rzeczywistym. Chcąc to uczynić należy w pierwszej kolejności opracować, bądź wybrać gotowy zbiór danych, na którym zostanie wykonane uczenie. Kluczowym elementem będzie zbudowanie modelu, który będzie jak najbardziej trafny. Celność predykcji jest kluczowa, aby usatysfakcjonować użytkownika. Metoda wykorzystania splotowych sieci neuronowych zostanie poddana wątpliwością. Przebadana zostanie teza, iż stworzony model oparty o wybrane dane będzie skuteczny w większości przypadków.

W celu zrozumienia celów oraz działania tych badań, zdefiniowano najważniejsze pojęcia, do których będą częste odwołania.

C. Sieć neuronowa

Sztuczna sieć neuronowa to rodzaj modelu uczenia się maszynowego, który jest zaprojektowany tak, aby działał podobnie jak biologiczna sieć neuronowa reprezentująca mózg zwierzęcia. Modele te są wykorzystywane do rozpoznawania złożonych wzorów i zależności, które istnieją w oznaczonym zbiorze danych. Przykładem może być zadanie identyfikacji obrazów zawierających jakiekolwiek dane, na przykład konkretny przedmiot czy zwierzę. Można to zrobić, analizując obrazy, oznaczone jako "pies" lub "brak psa" i wykorzystując wyniki do identyfikacji psów w innych obrazach. Sieć neuronowa znajdzie te zdjęcia w inny sposób niż wola ludzka. Wyciąga ona z materiału do nauki swój własny zestaw istotnych cech charakterystycznych.

Główna architektura modelu sieci neuronowej składa się z dużej liczby prostych węzłów przetwarzania danych zwanych neuronami, które są wzajemnie połączone i zorganizowane w różnych warstwach. Pojedynczy węzeł w warstwie jest połączony z kilkoma innymi węzłami w poprzedniej i następnej warstwie. Wejścia z jednej warstwy są odbierane i przetwarzane w celu wygenerowania sygnału wyjściowego, który jest przekazywany do następnej warstwy.

Pierwsza warstwa tej architektury jest często nazywana warstwą wejściową, która akceptuje parametry wejściowe. Ostatnia warstwa jest nazywana warstwą wyjściową, która zwraca parametry wyjście. Każda inna warstwa pomiędzy warstwą wejściową i wyjściową jest warstwą ukrytą.

D. Konwolucyjna sieć neuronowa

Konwolucyjne sieci neuronowe (z ang. Convolutional Neural Networks, w skrócie CNN) są przykładem głębokich sieci neuronowych. Głębokie sieci neuronowe składają się ze złożonych i wielu ukrytych warstw, które starają się wydobyć cechy z obrazów. W CNN, każde wejście obrazu jest traktowane jako macierz wartości pikseli, która reprezentuje natężenie ciemności przy danym pikselu na zdjęciu. Obraz w CNN traktowany jest jako sieć jednowymiarowa, uwzględniając lokalizację pikseli i sąsiadów do klasyfikacji. Operacje splotu s między funkcjami x oraz w oznaczamy znakiem gwiazdki:

$$s(t) = (x * w)(t) \tag{1}$$

gdzie:
s - wynik operacji slotu
x - funkcja poddawana operacji splotu

Sieć trenowana jest na pewnym podzbiorze par wejście-wynik, czyli zbiorze treningowym. Jednak skuteczność sieci i algorytmu uczenia są testowane już na osobnym podzbiorze. Zbiór testowy sprawdza jak dobrze sieć radzi sobie na nowych danych, różniących się od tych, na których była trenowana.

E. Wydajność

W celu porównania otrzymanych wyników należy na początku zdefiniować kryteria oceny. W tym przypadku badana będzie precyzja oraz błędy predykcji w zależności od ilości treningów klasyfikatora.

Precyzja niesie ze sobą informacje, jak dokładnie stworzony model sprawdza się w przewidywaniu wyrażeń. Z uwagi na fakt, iż występującym tu problemem jest klasyfikacja wieloklasowa, macierz błędów pomoże ustalić, które klasy są bardziej dominujące w stosunku do innych lub w stosunku do których klas model jest bardziej stronniczy. W ten sposób otrzymamy jasny obraz przewidywanego wyniku modelu.

Metoda uczenia nadzorowanego przebiega poprzez przekazanie zbioru uczącego do algorytmu. Następnie na

bazie macierzy wag przewiduje, do której klasy należą przykłady. Oblicza się wówczas błąd pomiędzy wartością zwróconą przez algorytm, a rzeczywistą wartością pochodzącą ze zbioru danych. Rachunek błędu wykonywany jest przy pomocy funkcji straty (ang. loss function). Celem jest zamiana parametrów modelu, aby zminimalizować funkcję strat na zbiorze treningowym.

II. Metoda

A. Wykorzystany sprzęt oraz oprogramowanie

Uczenie głębokich sieci neuronowych wymaga silnych jednostek obliczeniowych. Podczas obliczeń wykorzystano komputer wyposażony w procesor Intel Core i3-8350K, kartę graficzną GeForce GTX1060 z pamięcią 6GB oraz pamięć RAM 8 GB. Jeżeli chodzi o wymagania sprzętowe dla kamery, to wystarczy standardowa wbudowana w laptopie, bądź dokupiona i podłączona do komputera.

Do obliczeń wykorzystano język Python 3.6. W usprawnieniu badań użyto biblioteki open-source posiadające gotowe implementacje przydatnych dla nas funkcji.
Szczególnie można wyróżnić dwie. Są to OpenCV oraz Keras. OpenCV jest biblioteką szeroko stosowaną w przetwarzaniu obrazów, szczególnie obrazów w czasie rzeczywistym.

Natomiast Keras służy do szybkiego prototypowania architektur sieci neuronowych.

B. Dane

Do tych badań wykorzystana zostanie baza twarzy FER2013 (skrót z angielskiego Face Expression Recognition). Jest to zestaw 35 887 zdjęć osób, które wyrażają 7 emocji. Wszystkie obrazy są w skali szarości i mają rozdzielczość 48 na 48 pikseli. Obrazy są małe, więc jest małe wejście do modelu, a co za tym idzie, należy dostosowywać rozmiar ujęć z obrazu video o wysokiej rozdzielczości do niższej. Zatem rzutuje to na utratę szczegółów o wysokiej rozdzielczości. Niewielki uśmiech w takim przypadku nie zostanie zarejestrowany. Plik w formacie csv zawiera zbiór pikseli odpowiadający zdjęciu, sklasyfikowaną emocję oraz podział na próbki treningowe, walidacyjne i testowe, lecz dokonaliśmy podziału raz jeszcze. Zadaniem sieci neuronowej jest skategoryzowanie każdej twarzy w oparciu o emocje pokazane w wyrazie twarzy w jednej z siedmiu kategorii, gdzie:

- 0 = złość,
- 1 = zniesmaczenie,
- 2 = strach,
- 3 = szczęśliwy,
- 4 = smutek,
- 5 = zaskoczenie,
- 6 = obojętność.

Zestaw danych stanowi wyzwanie, ponieważ przedstawione twarze różnią się znacznie pod względem wieku i płci, pozycji twarzy oraz innych czynników, odzwierciedlających realistyczne warunki. Smutek i zniesmaczenie jest ciężkie do odróżnienia ze względu na podobne próbki. W internecie można spotkać opinię, iż nie jest to idealnie sklasyfikowany zbiór. Niemniej jednak w porównaniu z innymi zbiorami jest darmowy oraz ma sporą wielkość, dlatego więc zostanie wykorzystany.

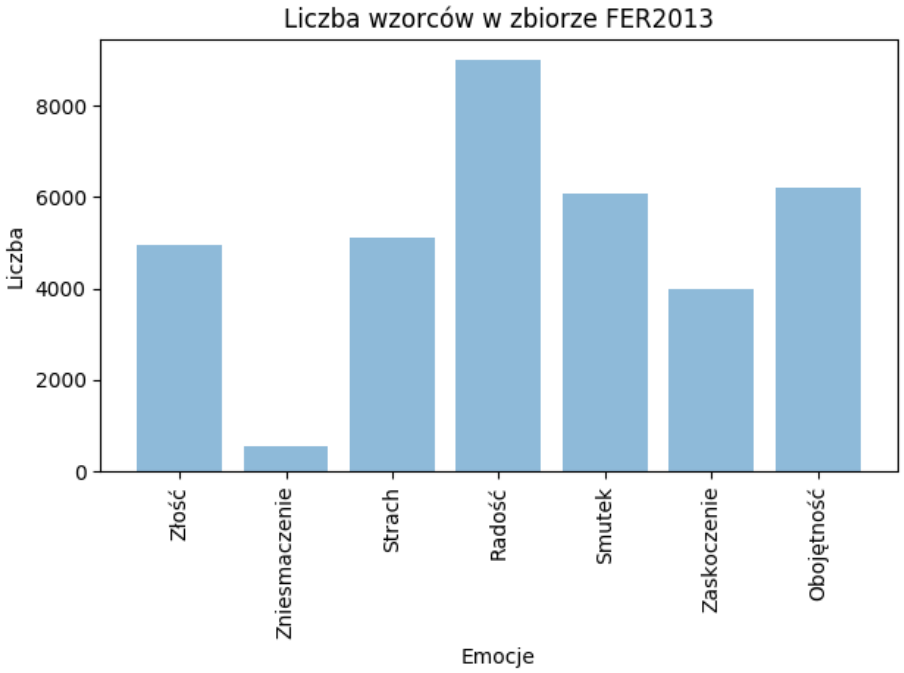
Rysunek 1: Przykładowe dane znajdujące się w bazie FER2013.

C. Model

Warstwa wyjściowa będzie się składać z 7 perceptronów, zgodnie z 7 różnymi etykietami emocji. Warstwa konwolucyjna, czasami nazywana mapą cech (ang. features map) stworzona jest z jednostek analizujących niewielki obszar danych wejściowych, np. 3x3 pikseli, tak jak w zastosowanym przez nas przypadku. Typowe wartości dla tego obszaru obejmują: (1, 1), (3, 3), (5, 5), (7, 7). Jeśli obrazy wejściowe są mniejsze niż 128×128, wówczas przyjmuje się rozmiar 3x3.

Pooling layer ma za zadanie uogólnić informacje dotyczące danej cechy wydobyte przez obszar jednostek warstwy konwolucyjnej, a co za tym idzie rozwiązywać problem nadmiarowości. Połączenie warstw konwolucyjnych oraz pooling pozwala na wydobycie kluczowych informacji zawartych w danych.

Pomiędzy warstwami konwolucyjnymi i warstwą gęstą (dense) znajduje się spłaszczona warstwa (flatten). Spłaszczenie służy jako połączenie między warstwami konwolucyjnymi i gęstymi. Dense to rodzaj warstwy, którą będziemy stosować w naszej warstwie wyjściowej. Pomiędzy nimi umieszczono również warstwę batch normalization oraz dropout. Pierwsza z nich normalizuje dane wyjściowe poprzedniej warstwy aktywacyjnej, odejmując średnią partii i dzieląc przez odchylenie standardowe partii, co przyspiesza proces szkolenia. Dropout natomiast losowo wyłącza kilka neuronów w sieci, aby zapobiec przeładowaniu.



Rysunek 2: Podział zbioru w zależności od emocji.

Będziemy mieli 7 węzłów w naszej warstwie wyjściowej, po jednym dla każdego możliwego wyniku (0-6). Za funkcję aktywacji przyjęto softmax, która wprowadza liczby reprezentujące prawdopodobieństwa. Wartość każdej mieści się w zakresie od 0 do 1 prawidłowego zakresu wartości. Rozkłady prawdopodobieństwa listy potencjalnych wyników prezentowane są jako wektor. Cały wektor wyjściowy sumuje się do 1. Predykcja wykonywana jest na podstawie tego, która opcja ma największą wartość. System zdrowia, którego częścią jest badanie emocji daje użytkownikowi pełną informację na temat uczuć. W związku z tym klasyfikacja przystępuje dla emocji o największym procencie prawdopodobieństwa, lecz pozostałe wartości są również wyświetlone.

D. Treninig

Do treningu CNN zostanie użyta tylko część wszystkich dostępnych próbek. Reszta będzie służyć jako zestaw testowy. Cały zestaw danych został podzielony w tym celu na dwie części. Zestaw szkoleniowy zawiera 70% obrazów, a zestaw testowy pozostałe 30%.

Przed przystąpieniem do treningu jest kilka ważnych parametrów, które trzeba określić, a są to:

- batch_size - rozmiar podzbioru przykładów do wykorzystania podczas jednej iteracji. Ponieważ jedna interacja jest zbyt wielka, aby uczyć na raz wszystkimi danymi, dlatego dzielimy ją na kilka mniejszych partii po 128 elementów
- image_height - wysokość przykładowych obrazów
- image_width - szerokość przykładowych obrazów
- channels - Liczba kolorów kanałów w przykładowym zdjęciu. W przypadku obrazów kolorowych liczba kanałów wynosi 3 (czerwony, zielony, niebieski). Dla obrazów monochromatycznych, tak jak w naszym przypadku jest tylko 1 kanał.
- epoches - liczba iteracji trenowania na całym zbiorze. Ilość iteracji nie jest jasno określona. Dla różnych zbiorów liczba iteracji jest związana z tym, jak różnorodne są dane.

Nasze dane będą przechodzić przez model 50 razy i partiami po 128 obrazów. Wykorzystamy dane testowe, aby zweryfikować model po każdej iteracji.

III. Plan eksperymentów

A. Wstępne przetwarzanie danych

Sieć neuronowa przyjmuje na wejście zdjęcie o rozmiarze 48x48 w odcieniach szarości. Następnie przeskalowuje wartości pikseli tak, aby zawierały się w przedziale od 0 do 1. Następnym krokiem po przygotowaniu zbioru danych jest wstępne przetwarzanie obrazu. Na każdym zdjęciu widzimy nie tylko twarz, ale całą głowę z małą częścią tułowia i tłem. Dla wyrazu twarzy te cechy są nieistotne i muszą być odcięte. Można to zrobić przy pomocy wbudowanych narzędzi z biblioteki OpenCV.

OpenCV zawiera szereg zaawansowanych klasyfikatorów do ogólnego wykrywania obiektów. Najpowszechniej znanym detektorem jest kaskada detektorów funkcyjnych Haar - opartych na detekcji twarzy. [10] Ten klasyfikator

podsumowuje intensywność pikseli w małych regionach obrazu, jak również wychwytuje różnicę pomiędzy sąsiednimi regionami obrazu. Został wyszkolony na wielu pozytywnych obrazach, czyli takich, na których występowała twarz, jak również negatywnych bez jakiejkolwiek lica. [11]–[13] Do tego projektu wykorzystano wstępnie przeszkolony klasyfikator do wykrywania twarzy z przodu. Plik nazywa się haarcascade_frontalface_default.xml.

B. Inicjacja sieci neuronowej

Typ modelu, którego będziemy używać to z ang. sequential. Jest to najprostszy sposób na zbudowanie modelu w Keras. Pozwala na budowanie modelu warstwa po warstwie.

W celu dodania warstwy splotu, wywołujemy funkcję add z obiektem klasyfikatora i przekazujemy Convolution2D (przeznaczona dla obrazów) z trzema parametrami. Pierwszym argumentem jest liczba detektorów funkcji, które chcemy utworzyć. Drugi i trzeci parametr to wymiary matrycy detektora. Powszechną praktyką jest rozpoczynanie od 32 detektorów funkcji dla CNN. Następnym parametrem jest input_shape, czyli kształt obrazu wejściowego. Obrazy zostaną przekształcone w ten kształt podczas przetwarzania wstępnego. Ostatnim parametrem jest funkcja aktywacji. Klasyfikowanie obrazów jest problemem nieliniowym. Wobec tego używamy funkcji prostownika, aby upewnić się, że nie mamy ujemnych wartości pikseli podczas obliczeń. W ten sposób osiągamy nieliniowość.

W tym kroku zmniejszamy rozmiar mapy obiektów. Tworzymy pulę o wielkości 2x2, by zmniejszyć rozmiar mapy obiektów bez utraty ważnych informacji o obrazie. Wszystkie połączone mapy obiektów są pobierane i umieszczane w jednym wektorze. Funkcja Flatten spłaszcza wszystkie mapy funkcji w jednej kolumnie. Następnym krokiem jest użycie wektora, który otrzymaliśmy powyżej, jako wejścia do sieci neuronowej przy użyciu funkcji Dense w Keras. Jej parametrem jest output_dim, czyli liczba węzłów w ukrytej warstwie. Częstą praktyką jest wybieranie liczby węzłów w potęgach dwójki. Kolejnym krokiem jest funkcja aktywacji. Zwykle używa się funkcji ReLu w ukrytej warstwie.

Następną warstwą, którą musimy dodać, jest warstwa wyjściowa. Na koniec parametr funkcji Dense przybiera wartość ilości oczekiwanych klas, a więc w tym wypadku 7.

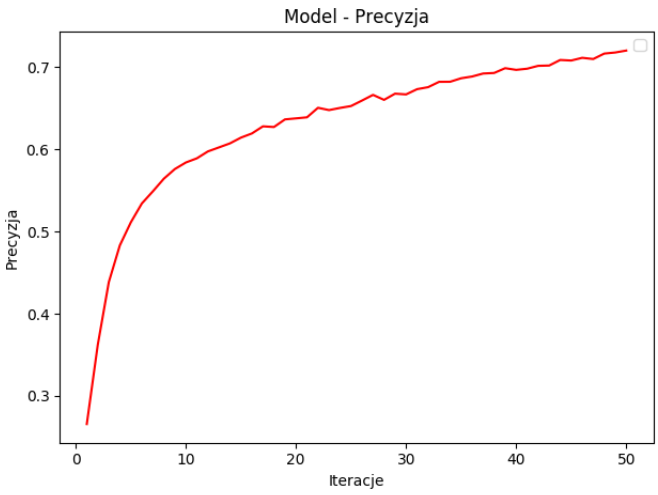
Ostatnim krokiem jest funkcja kompilacji. Przyjmuje ona trzy parametry: optymalizator, funkcję utraty i metrykę wydajności. Optymalizator to algorytm gradientu spadku, którego będziemy używać. Korzystamy z funkcji straty categorical_crossentropy, ponieważ wykonujemy klasyfikację wieloklasową.

Warstwa	Wejście	Wyjście
conv2d_1	46, 46, 64	46, 46, 64
conv2d_2	46, 46, 64	46, 46, 64
batch_normalization_1	46, 46, 64	46, 46, 64
max_pooling2d_1	46, 46, 64	23, 23, 64
dropout_1	23, 23, 64	23, 23, 64
conv2d_3	23, 23, 64	23, 23, 128
conv2d_4	23, 23, 128	23, 23, 128
batch_normalization_2	23, 23, 128	23, 23, 128
max_pooling2d_2	23, 23, 128	11, 11, 128
dropout_2	11, 11, 128	11, 11, 128
conv2d_5	11, 11, 256	11, 11, 256
conv2d_6	11, 11, 256	11, 11, 256
batch_normalization_3	11, 11, 256	11, 11, 256
max_pooling2d_3	5, 5, 256	5, 5, 256
dropout_3	5, 5, 256	5, 5, 256
flatten_1	5, 5, 256	6400
dense_1	6400	256
batch_normalization_4	256	256
activation_1	256	256
dropout_4	256	256
dense_2	256	128
batch_normalization_5	128	128
activation_2	128	128
dropout_5	128	128
dense_3	128	7

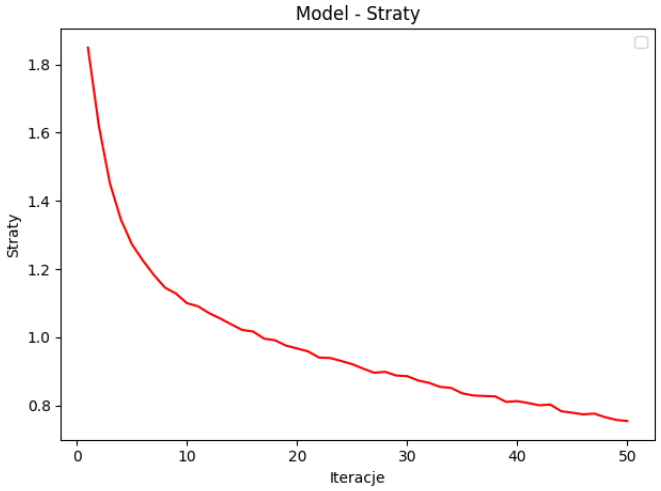
Tabela I: Architektura zaprojektowanej sieci

IV. Badania

Przeprowadzone badania na temat trenowania sieci splotowej zostały sporządzone na dwóch wykresach. Pierwszy z nich mówi o tym, w jaki sposób zachowuje się precyzja predykcji w zależności od ilości iteracji nauczania. Wykres na rysunku nr 4. przedstawia zaś w ten sam sposób informacje na temat funkcji strat. Informuje ona o ilości popełnianych błędów przez nasz model, dlatego jest istotnym miernikiem wartości modelu.

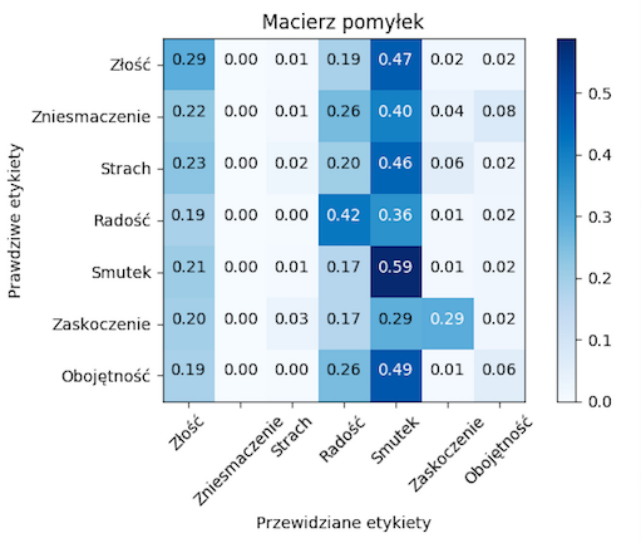


Rysunek 3: Wykres przedstawiający wzrost precyzji wraz ze zwiększeniem ilości iteracji.



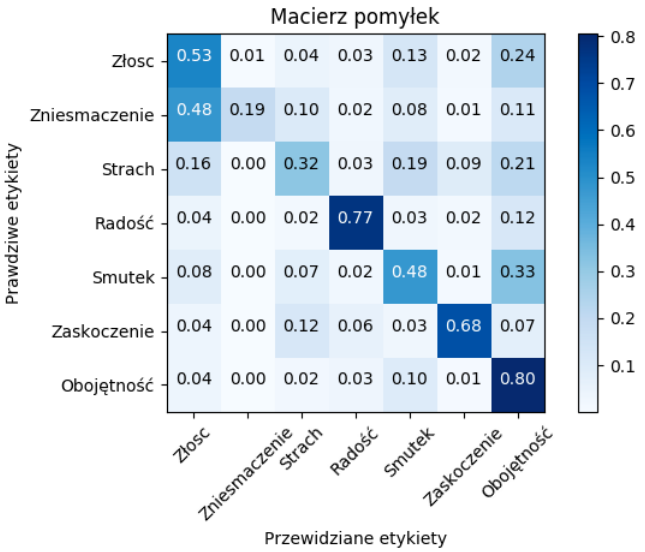
Rysunek 4: Wykres przedstawiający spadek strat wraz ze zwiększeniem ilości iteracji.

Podczas określania parametrów niezbędnych do przeprowadzenia nauczania, zdecydowaliśmy się na 50 iteracji. Niemniej jednak, nie po każdej iteracji model został zapisany do pliku o rozszerzeniu hdf5. Zachowane zostały tylko te, które uzyskały lepszy wyniki niż poprzednie. Ostatni zapisany model osiągnął precyzję 66% dla 30 iteracji.



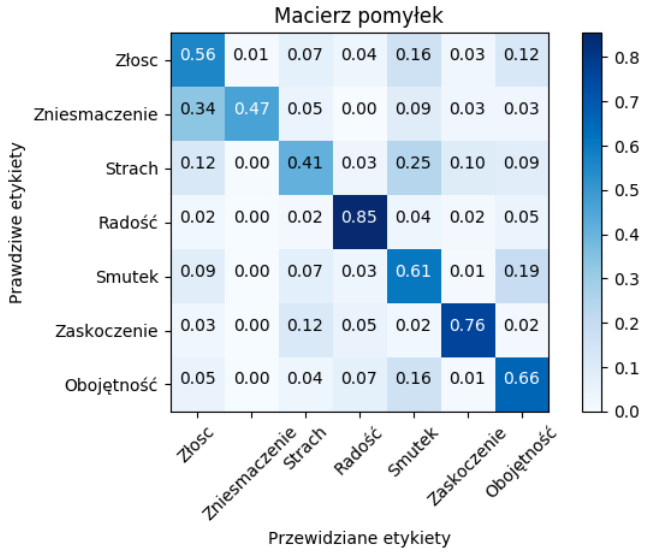
Rysunek 5: Macierz błędów dla 1. iteracji.

Rysunek 5., przedstawiający macierz pomyłek dla pierwszej iteracji zdecydowanie pozostawia wiele do życzenia. Chociaż smutek został sklasyfikowany najlepiej, bo w 59% poprawnie, to niepokojące jest zestawienie przykładowo dla złości. Dane procentowe przedstawiają nieprawidłowe klasyfikacje dla ponad połowy emocji. Złość została błędnie odczytana jako smutek w 47% przypadków, a prawidłowo tylko w 29%. Warto również zauważyć brak sklasyfikowania zniesmaczenia.



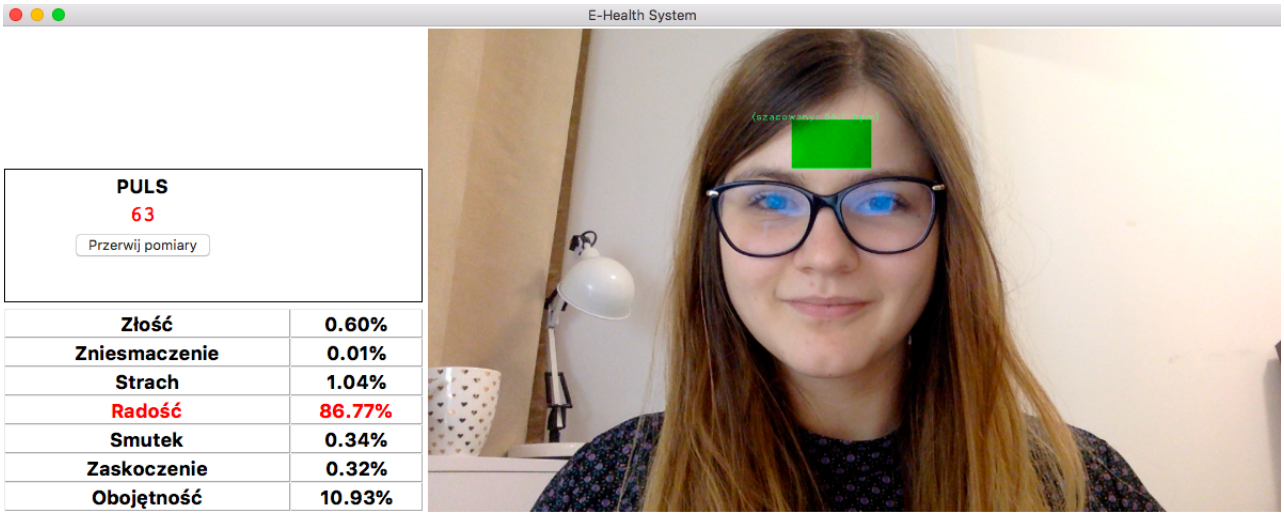
Rysunek 6: Macierz błędów dla 15. iteracji.

W połowie treningu, po 15 iteracjach, otrzymaliśmy dokładniejsze wyniki. Zdecydowanie widać, iż najlepiej sklasyfikowana jest obojętność, a zaraz za nią radość. Zniesmaczenie jest trudne do rozróżnienia ze złością, a smutek wielokrotnie został ujęty jako zobojętnienie. Podczas dalszego treningu mieliśmy nadzieję na zmniejszenie błędnych klasyfikacji, a w szczególności uzyskanie lepszych wyników dla zniesmaczenia.



Rysunek 7: Macierz błędów dla 30. iteracji.

Rezultat 30 serii treningów przedstawia się lepiej niż w poprzednim przypadku. Widać też znaczącą różnicą między 1, a 30 epoką. Pięć na siedem emocji zostało sklasyfikowanych chociaż w połowie przypadków poprawnie. Obecne wyniki między zniesmaczeniem, a złością przedstawiają się korzystniej na rzecz pierwszej z emocji, lecz różnice



Rysunek 8: Zrzut ekranu okna programu wykorzystującego zaproponowany model po 30 iteracjach.

wciąż nie są spore. Warto zwrócić uwagę jak kształtują się wyniki, porównując je z zawartością zbioru zdjęć. Ilustracji sklasyfikowanych jako radość jest 16 razy więcej niż w przypadku zniesmaczenia. Większe odchylenia w predykcji występują między emocjami podobnymi i często trudnymi do odróżnienia nawet dla ludzi, czyli strachu i smutku oraz smutku i zobojętnienia. W przypadku przeciwstawnych uczuć jak radość i smutek błąd jest stosunkowo niewielki.

Poza badaniami statystycznymi, przeprowadzono również próby detekcji emocji w czasie rzeczywistym poprzez strumień wideo z kamerki w laptopie. Nie da się zaprzeczyć, że emocje o największym dopasowaniu są zdecydowanie łatwiejsze do wykrycia przez program. Zniesmaczony wyraz twarzy jest bardzo trudny do uzyskania, ponieważ tak jak przedstawia to macierz pomyłek, jest często mylona ze złością.

V. Podsumowanie

Analiza wyników badań daje do zrozumienia, iż zwiększenie iteracji może wpływać pozytywnie na naukę modelu. Niemniej jednak w końcu dochodzi się do miejsca, w którym dalszy trening nie wpływa już korzystnie, a wyniki wahają się. Celem przeprowadzonych obserwacji było znalezienie odpowiedzi na pytanie, czy za pomocą splotowych sieci neuronowych możemy zbudować taki model, który byłby wstanie dobrze rozpoznawać ludzkie emocje. Nie jesteśmy w stanie jasno stwierdzić, czy uzyskane wyniki potwierdzają, bądź zaprzeczają tej teorii. Precyzja na poziomie 66% daje niewiele ponad połowę poprawnie sklasyfikowanych przypadków. Możemy jednak wziąć pod uwagę fakt, iż zwycięzca konkursu ogłoszonego przez Kaggle uzyskał na tym samym zbiorze danych FER2013 wynik 71%. Przypuszczalnie modyfikacja architektury modelu mogłaby pomóc w osiągnięciu jeszcze lepszych wyników.

Literatura

[1] Wim Verkruysse, Lars O Svaasand, J Stuart Nelson, “Remote plethysmographic imaging using ambient light, Uniwersytet Kalifornijski, 2008
[2] Abir Fathallah,“Facial Expression Recognition via Deep Learning,’2017
[3] Erick Cambria, “Affective Computing and Sentiment Analysis,” 2016
[4] Arushi Raghuvanshi,“Facial Expression Recognition with Convolutional Neural Networks,”, Uniwerstytet Stanforda, 2016
[5] Zhiding Yu Carnegie, “Image based Static Facial Expression Recognition with Multiple Deep Network Learning, Uniwerstytet Carnegie Mellon, 2015
[6] Jinwoo Jeon, Jun-Cheol Park, YoungJoo Jo, ChangMo Nam, Kyung-Hoon Bae, Youngkyoo Hwang, Dae-Shik Kim,“A Real-time Facial Expression Recognizer using Deep Neural Network,”2016
[7] Christopher Pramerdorfer, Martin Kampel,“Facial Expression Recognition using Convolutional Neural Networks: State of the Art,”, Computer Vision Lab, Uniwersytet Techniczny w Wiedniu, Austria, 2016
[8] Shan Li, Weihong Deng,“Deep Facial Expression Recognition: A Survey, ” 2018
[9] Asim Jan, “Deep Learning based facial expression recognition and its aplications,” Uniwerstytet Burnel w Londynie, 2017
[10] Haoxiang Li, “A Convolutional Neural Network Cascade for Face Detection,” 2015
[11] Phillip Ian Wilson,“Facial feature detection using Haar Classifiers,” 2006
[12] Asif Anjum Akash, Abdus Salim Mollah, Akhand MAH,“Improvement of Haar Feature Based Face Detection in OpenCV Incorporating Human Skin Color Characteristic”, 2016
[13] Li Z., Xue L., Tan F., “Face Detection in Complex Background Based on Skin Color Features and Improved Adaboost Algorithm,” 2010