

**WYDZIAŁ ZARZĄDZANIA**

Katedra Informatyki Biznesowej i Inżynierii Zarządzania

Projekt dyplomowy

*Aplikacja mobilna do nauki matematyki dla szkoły podstawowej z wykorzystaniem machine learningu do rozpoznawania odpowiedzi pisanych ręcznie*

*A mobile math learning app for elementary school using machine learning to recognize handwritten answers*

Autor: *Monika Helena Etrych*

Kierunek studiów: Informatyka i Ekonometria

Opiekun pracy: *dr Beata Basiura*

Kraków, 2023 r.

Spis treści

[Spis rysunków 3](#_Toc131054341)

[Spis tabel 3](#_Toc131054342)

[Wstęp 4](#_Toc131054343)

[1. Rozpoznawanie cyfr i ich interpretacja 5](#_Toc131054344)

[1.1. Interpretacja obrazów przez komputer 5](#_Toc131054345)

[1.2. Metody uczenia maszynowego do rozpoznawania cyfr 6](#_Toc131054346)

[1.3. Obróbka obrazów z cyframi 7](#_Toc131054347)

[1.3.1. Wstępne przetwarzanie obrazów 7](#_Toc131054348)

[1.3.2. Augmentacja 8](#_Toc131054349)

[1.4. Weryfikacja modelu do rozpoznawania cyfr 9](#_Toc131054350)

[2. Model aplikacji do nauki uczenia matematyki 12](#_Toc131054351)

[2.1. Założenia modelu aplikacji 12](#_Toc131054352)

[2.2. Wybrane technologie zastosowane do budowy aplikacji 13](#_Toc131054353)

[2.3. Wybrane metody uczenia maszynowego zastosowane w aplikacji 14](#_Toc131054354)

[2.4. Korzystanie z modelu ML poprzez API HTTP 16](#_Toc131054355)

[3. Opis implementacji i działania aplikacji 17](#_Toc131054356)

[3.1. Ekran startowy i podstawowe funkcjonalności 17](#_Toc131054357)

[3.2. Schematy, opisy poszczególnych funkcjonalności, opis kodów 18](#_Toc131054358)

[3.3. Przykłady zastosowania aplikacji w konkretnych zadaniach 19](#_Toc131054359)

[Zakończenie 20](#_Toc131054360)

[Spis tabel, wykresów i rysunków 21](#_Toc131054361)

[Wykaz rysunków 21](#_Toc131054362)

[Bibliografia 22](#_Toc131054363)

[Aneks (kod, tabelka z danymi, treść ankiety) 24](#_Toc131054364)

# Spis rysunków

[Rysunek 1‑1 Cztery reprezentacje 5](#_Toc131054303)

[Rysunek 1‑2 Losowy obrót obrazów 8](#_Toc131054304)

[Rysunek 1‑3 Macierz błędu 10](#_Toc131054305)

# Spis tabel

# Wstęp

# Rozpoznawanie cyfr i ich interpretacja

## Interpretacja obrazów przez komputer

Obrazy reprezentowane przez komputer składają z pikseli - kwadratów, które posiadają kilka wartości określających kolor. Wyróżnia się 4 podstawowe reprezentacje: binarną, monochromatyczną (ang. grayscale), RGB i RGBA (rys. 1.1).

Reprezentacja binarna oraz monochromatyczna składają się tylko z jednej wartości dla danego piksela. W przypadku reprezentacji binarnej wartość jest zerojedynkowa, gdzie 0 oznacza, że pixel ma kolor czarny, natomiast 1 to kolor biały. W reprezentacji monochromatycznej wartość jest z zakresu od 0 do 255. monochromatycznej tym przypadku 0 oznacza kolor czarny, 255 - biały, natomiast wartości pośrednie, to odcienie szarości.

RGB zawiera trzy wartości określające kolor: czerwony, zielony i niebieski. RGBA posiada dodatkowo jedną wartość, która określa poziom przezroczystości. [1]

Rysunek 1.1 Cztery podstawowe reprezentacje pikseli



źródło:[https://maxcandocia.com/article/2016/Apr/06/how-computers-recognize- images/](https://maxcandocia.com/article/2016/Apr/06/how-computers-recognize-%20images/)

UWAGA : W reprezentacji i rozpoznawaniu obrazów dużą rolę pełni kontrast. (tutaj kilka słów o  kolorach i zaburzeniach i reprezentacji kontrastu)! Kilka słów też o rozmiarach i kodowaniu danych ich reprezentacji !

## Metody uczenia maszynowego do rozpoznawania cyfr

Metody uczenia maszynowego są coraz częściej wykorzystywane do rozpoznawania cyfr w różnych dziedzinach, takich jak bankowość czy usługi pocztowe. Uczenie maszynowe pozwala na automatyczne i precyzyjne rozpoznawanie cyfr. Wiele papierowych formularzy w  sposób szybki i dokładny może zostać przeniesiona do komputera dzięki takim możliwościom. W usługach pocztowych natomiast rozpoznawanie kodów pocztowych znacząco przyspiesza proces sortowania listów i przesyłek [2].

Do rozpoznawania cyfr wykorzystuje się algorytmy uczenia nadzorowanego. W tym podejściu, dane uczące zawierają cyfry wraz z odpowiadającymi im etykietami, czyli poprawnymi wartościami cyfr. Algorytm uczony na takich danych jest w stanie nauczyć się rozpoznawać cyfry na podstawie cech wizualnych, takich jak kształt i proporcje. [1, 3]

Wśród popularnych algorytmów uczenia maszynowego stosowanych do rozpoznawania cyfr znajdują się między innymi konwolucyjne sieci neuronowe (CNN), wielowarstwowe sieci neuronowe (MLP) oraz algorytm maszyny? wektorów nośnych (SVM). Zastosowanie tych algorytmów wraz z odpowiednio przygotowanymi danymi uczącymi, pozwala na uzyskanie skuteczności rozpoznawania cyfr na poziomie nawet 99% [3, 4].

Algorytm maszyny wektorów nośnych polega na maksymalizacji odległości pomiędzy hiperpłaszczyzną, a marginesem przynależności do poszczególnych klas, poprzez zmianę położenia hiperpłaszczyzny. Istnieje wiele możliwych hiperpłaszczyzn, a jej kształt zależy od złożoności problemu - liczby klas do zaklasyfikowania. Kiedy istnieją dwie klasy, to hiperpłaszczyzna jest prostą, dla trzech danych klas wejściowych jest płaszczyzną 2D itd. [5].

Wielowarstwowy perceptron w sieci neuronowej składa się z trzech podstawowych warstw: warstwy wejściowej, warstw ukrytych oraz warstwy wyjściowej. Liczba warstw ukrytych zależy od złożoności problemu. Warstwy składają się z neuronów oraz funkcji aktywacji. Wraz z kolejnymi iteracjami modelu, wagi są aktualizowane tak, aby zminimalizować błąd predykcji. Wagi te określają poziom wpływu każdego neuronu na wynik sieci. Funkcje aktywacji pozwalają na nieliniowe modelowanie złożonych zależności, co z kolei wpływa na lepszą predykcję [6].

W konwolucyjnych sieciach neuronowych można wyróżnić sześć warstw: wejściową, konwolucyjną, aktywacji, łączenia, w pełni połączoną oraz wyjściową. Warstwa konwolucyjna składa się z filtrów i map cech. Filtry są to kwadratowe wycinki obrazu, które wyodrębniają cechy charakterystyczne dla różnych obrazów. Mapa cech jest zbiorem wyodrębnionych cech. Warstwa aktywacji przekształca dane z mapy cech tak aby dodać nieliniowość. Zwykle używa wybranej funkcji np. ReLU. Warstwa łączenia (ang. pooling) generalizuje cechy, aby zapobiec przeuczeniu. Warstwa w pełni połączona jest używana do stworzenia końcowej nieliniowej kombinacji cech oraz dokonywania przewidywań przez sieć [3, 6, 7].

Wymienione (dwie?) metody zostały porównane w 2021r przez zespół R. Dixita. W  procesie wstępnego przetwarzania obrazu wykorzystano tylko podstawowe przekształcenia takie jak normalizacja wartości pikseli, co pozwoliło na zamianę wartości pikseli z przedziału 0-255 na wartości z przedziału 0-1. Ponadto, wartość liczby przedstawionej/prezentowanej na obrazie została zamieniona na zmienną kategoryczną. Następnie wybrane metody zostały zestawione pod kątem czasu wykonania oraz dokładności dla danych treningowych i  testowych. Najwyższy wynik dokładności dla danych treningowych osiągnęła metoda SVM - 99,98%, natomiast dla danych testowych – sieci CNN - 99,31% [3].

W innym badaniu z roku 2012r przeprowadzonym przez zespół D.C. Ciresana porównano duże modele wielowarstwowe sieci neuronowych (MLP) z różnymi konfiguracjami liczby warstw oraz neuronów. W rezultacie najlepszy wynik rozpoznawania obrazów cyfr uzyskano dla MLP składającej się z ośmiu warstw i liczb neuronów w warstwach ukrytych ??: 2500, 2000, 1500, 1000, 500, 10. Proces uczenia modelu obejmował aż 2000 epok, a osiągnięta dokładność wyniosła 99,65%. Warto zauważyć, że cyfry, które zostały źle sklasyfikowane przez model, były trudne do określenia nawet dla człowieka [8].

W 2020r zespół S. Ahlawata podjął próbę dopasowania hiperparametrów do modelu konwolucyjnych sieci neuronowych. W przeprowadzonym badaniu sprawdzono wpływ liczby warstw, rozmiaru kroku (liczby pixeli o jaką przesuwa się filtr), pól recepcyjnych (wycinka obrazu odpowiadającego danej cesze), rozmiaru filtrów (rozmiaru macierzy z wagami), dopełnienia (paddingu), rozcieńczenia (dilution) oraz rodzajów klasyfikatorów. W ramach badań zastosowano kilka metod przygotowania danych takie jak: skalowanie, centrowanie, redukcje szumów. Najlepszy wynik, który udało się osiągnąć to dokładność na poziomie 99,89% używając optymalizatora Adam [9].

Ten optymalizator może trochę opisać ? Na czym polega taka optymalizacja?

## Obróbka obrazów z cyframi

### Wstępne przetwarzanie obrazów

W procesie rozpoznawania cyfr przez model, kluczowe znaczenie ma odpowiednie przekształcenie danych reprezentujących obraz przed ich przekazaniem do modelu. Wstępne przetwarzanie obrazów ma na celu wymagane przygotowanie danych do dalszej analizy. Ponadto zmniejsza czas potrzebny do wytrenowania modelu i zwiększa szybkość otrzymania wyniku. Na przykład, kiedy wejściowy obraz jest stosunkowo duży, zmniejszenie rozmiaru macierzy danych znacząco zmniejsza czas trenowania, bez znacznej utraty dokładności modelu.

Sieci neuronowe wymagają, aby dane wejściowe miały ten sam rozmiar (chodzi o rozmiar pliku graficznego/ czy samej reprezentacji?), ponieważ sieci te korzystają z macierzy wag, które mają ustaloną liczbę kolumn i wierszy. Przetwarzanie obrazów przez ustawienie stałych wymiarów liczby pikseli? jest jednym ze sposobów na dostosowanie rozmiaru obrazów do wymogów sieci neuronowej. Zmiany rozmiaru obrazu można dokonać na dwa sposoby. Pierwszy z nich polega na zmniejszeniu obrazu poprzez centrowanie oraz wycinanie pustego obszaru jeśli obraz jest za duży. Drugi sposób, stosowany w przypadku gdy obraz jest za mały, polega na zwiększeniu rozmiaru przez dodanie pustych pikseli po bokach.

Złożoność modelu można uprościć za pomocą zamiany reprezentacji obrazu na mniej złożoną. Dla kolorowych zdjęć stosuje się zamianę na skalę monochromatyczną, a dla skali szarości zamianę na reprezentację binarną [10]. W ten sposób oczywiście plik i oróbka danych się upraszacza ale czy nie spada efektywność rozpoznawania ?? Jedna gubimy tutaj jakieś dane ?

### Augmentacja

W sytuacji, kiedy danych treningowych jest zbyt mało, model głębokiego uczenia może ulec przeuczeniu, czyli zbytniemu dopasowaniu do danych trenujących. Przeuczenie skutkuje niskimi wynikami otrzymanymi na zbiorze testowy ? na nowych danych. Aby zapobiec temu problemowi, konieczne jest dostarczenie modelowi odpowiedniej liczby danych trenujących. Można to osiągnąć stosując augmentację, czyli przekształcenia istniejących obrazów za pomocą np. losowych zmian rotacji lub jasności. Na rysunku 1.1 w pierwszym rzędzie przedstawiono obrazy cyfr w normalnym położeniu oraz w drugim rzędzie pokazane zostały te same cyfry po zastosowaniu zmian rotacji o losowy kąt z zakresu ± 30 stopni. Augmentacja jest łatwym sposobem na zwiększenie różnorodności danych trenujących i  tym samym poprawę jakości modelu.

Innymi sposobami na zwiększenie liczby danych są losowe zmiany jasności i naświetlenia, czy dodawanie pikseli w losowych miejscach.

Rysunek 1.2 Losowy obrót obrazów

Obraz zawierający tekst

Opis wygenerowany automatycznie

*źródło: opracowanie własne*

Te same zabiegi przetwarzania danych mogą być stosowane zarówno w ramach wstępnego przetwarzania danych, jak i augmentacji. Jednakże, wstępne przetwarzanie danych jest stosowane zarówno na danych treningowych, jak i testowych, podczas gdy augmentacja jest stosowana wyłącznie na danych treningowych po wstępnym przetworzeniu [10].

## Weryfikacja modelu do rozpoznawania cyfr

Poziom dokładności stanowi jedną z najprostszych metod oceny jakości modelu. Określa on stosunek poprawnie sklasyfikowanych danych do liczby wszystkich danych. Miara ta jest adekwatna tylko, gdy zbiór danych jest zbilansowany, czyli zawiera równą liczbę próbek dla każdej klasy [11].

Innym sposobem weryfikacji modelu uczenia maszynowego jest sprawdzenie na jakim etapie nauczenia znajduje się tworzony model. Taki model pod wpływem czasu może być kolejno: niedouczony, wyuczony odpowiednio i przeuczony. Jednak, jeśli model jest źle skonstruowany może nie nigdy nie osiągnąć późniejszych etapów uczenia.

Problem niedouczenia polega na tym, że hipotezy modelu są nieodpowiednie i zbyt proste w porównaniu do złożoności danych. Przez to sam model nie opisuje/tłumaczy wystarczająco dobrze danych. Natomiast przeuczenie polega na zbytniej złożoności modelu i  w konsekwencji zbyt dokładnym odwzorowywaniu danych treningowych. Jest to przyczyną problemów w odpowiednim dopasowaniu nowych elementów [12]. Dopasowanie modelu do specyficznych zachowań zawartych w danych treningowych nie pozwala uogólnić wyników na nowe przypadki danych. UWAGA: A przyczyny tego stanu?

Dwa wykresy: wykres dokładności i wykres strat, mogą dostarczyć informacji o stanie wyuczeniu modelu. Funkcja dokładności przedstawia poziom dokładności modelu w kolejnych epokach uczenia. Tak długo jak obydwie te krzywe dla danych uczących i testujących rosną model polepsza swoją skuteczność ale model nie jest jeszcze wystarczająco wyuczony. Kiedy funkcje stabilizują się, nie rosną tak gwałtownie mamy model odpowiednio wyuczony. Jednak, gdy funkcja (która?) na zestawie danych testujących po pewnym czasie zacznie maleć, to oznacza, że model został przeuczony. Celem w uczeniu maszynowym jest maksymalizacja współczynnika dokładności i minimalizacja strat, błędu [13].

Podczas klasyfikacji można rozróżnić cztery przypadki zaklasyfikowania (rys. 1.3):

Rysunek 1.3 Macierz błędu

Obraz zawierający stół

Opis wygenerowany automatycznie

*źródło: opracowanie własne*

Wyróżnia się procentowe wskaźniki, które bazują na wskazanych powyżej (rys. 1.3) wartościach: czułość, specyficzność oraz precyzja. W zależności od oczekiwań danego modelu powinno się maksymalizować wybrany współczynnik. Nie jest możliwe otrzymanie wszystkich współczynników na wysokim poziomie lub na niskim poziomie.

Czułość jest to stosunek poprawnie zaklasyfikowanych wartości jako pozytywne (TP) do sumy wartości prawdziwie pozytywnych (TP) i fałszywie negatywnych (FN): TP/(TP+FN). Jest wykorzystywana, gdy niepożądane jest wystąpienie nieprawdziwie negatywnych wyników klasyfikacji.

Specyficzność to stosunek wyników prawdziwie negatywnych (TN) do sumy prawdziwie negatywnych (TN) i fałszywie pozytywnych (FP): TN/(TN+FP). Jest używana, kiedy ważne jest zwrócenie uwagi na prawdziwie negatywne przypadki.

Precyzja to stosunek prawdziwie pozytywnych (TP) do sumy prawdziwie pozytywnych (TP) i fałszywie pozytywnych (FP): TP/TP+FP. Znajduje ona zastosowanie, gdy ważne są prawdziwie pozytywne wyniki sklasyfikowania [14, 15].

A jak się to wyznacza gdy klasyfikacja jest wielo-kategoryczna ?? Czyli jak to będzie wyglądało w przypadku 10 cyfr ?

Czy można te wskaźniki zastosować do wyników sieci neuronowych czy też do sieci stosuje się inne wskaźniki?

UWAGI:

Pisze Pani bardzo ładnie – styl bardzo dobry , ale pisze Pani bardzo ogólnie. Trzeba to trochę rozwinąć pod kątem tematu pracy (może Pani to rozwinąć później jak będzie gotowa część praktyczna)

# Model aplikacji do nauki uczenia matematyki

## Założenia modelu aplikacji

## Wybrane technologie zastosowane do budowy aplikacji

## Wybrane metody uczenia maszynowego zastosowane w aplikacji

Augmentacja: rotate, zoom, shift, shear

<https://medium.com/the-data-science-publication/how-to-augment-the-mnist-dataset-using-tensorflow-4fbf113e99a0>

## Korzystanie z modelu ML poprzez API HTTP

# Opis implementacji i działania aplikacji

## Ekran startowy i podstawowe funkcjonalności

## Schematy, opisy poszczególnych funkcjonalności, opis kodów

## Przykłady zastosowania aplikacji w konkretnych zadaniach

# Zakończenie

Wyniki, rezultaty, udowodnienie tezy, co na przyszłość rozwinąć

# Bibliografia

[1] M. [Candocia](https://www.linkedin.com/in/max-candocia-a270b574/" \t "_blank)*, A Simple Explanation of how Computers Recognize Images,* [Analysis for Many Audiences,](https://maxcandocia.com/)  2016, url: <https://maxcandocia.com/article/2016/Apr/06/how-computers-recognize-images/>, dostęp: 23.02.2023

[2] S. M. Shamim, M. B. A. Miah, A. Sarker, M. Rana, A. A. Jobair, 2018, *Handwritten Digit Recognition Using Machine Learning,* Algorithms Indonesian Journal of Science & Technology (1) vol. 3, str.: 29-39 ISSN 2528-1410, ISSN 2527-8045

[3] R. Dixit, R. Kushwah, S. Pashine, *Handwritten Digit Recognition using Machine and  
Deep Learning Algorithms,* Artykuł Naukowy 2021, url: <https://arxiv.org/pdf/2106.12614.pdf>, dostęp: 1.03.2023

[4] M. Javed, The Best Machine Learning Algorithm for Handwritten Digits Recognition, 2020, url:<https://towardsdatascience.com/the-best-machine-learning-algorithm-for-handwritten-digits-recognition-2c6089ad8f09>, dostęp: 1.03.2023

[5] R. Gandhi, *Support Vector Machine — Introduction to Machine Learning Algorithms*, 2018, url: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>, dostęp: 1.03.2023

[6] J. Brownlee, *Develop Deep Learning Models on Theano and TensorFlow Using Keras,* 2016, str: 37-42*,* 116 – 134

[7] M. Mamczur, *Jak działają konwolucyjne sieci neuronowe (CNN)?,* 2021, url: <https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/>, dostęp: 14.03.2023

[8] D.C. Ciresan, U. Meier, L. M. Gambardella, J. Schmidhuber, *Deep Big Multilayer Perceptrons For Digit Recognition,* 2012, w: G. Montavon, G.B. Orr, KR. Müller, *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, vol 7700, str: 581-598, Springer, Berlin, Heidelberg

[9] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, B. Yoon, 2020, *Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)*

[10] J. Nelson*, What is Image Preprocessing and Augmentation?,* Roboflow, 2020, url: <https://blog.roboflow.com/why-preprocess-augment/>, dostęp: 28.02.2023

[11] N. Bressler, *How to Check the Accuracy of Your Machine Learning Model,* 2022, url: <https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model/>, dostęp: 23.03.2023

[12] D. Nikolaiev *Overfitting and Underfitting Principles,* 2021,url: *<https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c>,* dostęp: *14.03.2023*

[13] J. Brownlee, *How to use Learning Curves to Diagnose Machine Learning Model Performance, 2019,* url: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>, dostęp: 9.03.2023

[14] A Lekhtman, 2019, *Data Science in Medicine — Precision & Recall or Specificity & Sensitivity?,* url: <https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1>, dostęp: 29.03.2023

[15] S. Ghoneim, 2019, *Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?,* url: <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124>, dostęp: 29.03.2023

# Aneks (kod, tabelka z danymi, treść ankiety)