

# **Sprawozdanie**

## Inteligentna Analiza Danych

Konrad Pławik

data oddania zadania: 20 maja 2024

# 1 Cel zadania

Celem zadania było przetestowanie możliwości rozpoznawania znaków graficznych przez sieć neuronową typu MADALINE. W tym celu skorzystaliśmy z sieci napisanej na zajęciach z tą różnicą że predefiniowane znaki zastąpiliśmy możliwością odczytu ich z plików PNG. Do wygenerowania plików ze znakami skorzystaliśmy ze skryptu napisanego w języku Python i bibliotek PIL oraz OpenCV.

# 2 Wstęp teoretyczny

Zbudowana sieć wykorzystywała model neuronu, wzorca, zestawu wzorców oraz model sieci. Model sieci miał zaimplementowane metody treningu oraz testowania. Dodatkowo także utworzenia dodatkowych funkcji i metod pomocniczych, jak np. metody do wyświetlania zawartość zestawów wzorców oraz funkcje do działań na wektorach (iloczyn wektorowy, normalizacja, wyznaczanie długości).

W celu prostej automatyzacji stworzony został również skrypt tworzący obrazki z zadanymi literami wybranej czcionki o zadanym rozmiarze i stopniu zaszumienia. W głównym katalogu znajdują się skrypty powłoki bash (`create_trainset.sh` oraz `create_testsets.sh`) które uruchamiają w/w skrypt i generują zestawy danych: treningowy oraz testowe dla każdego z siedmiu przeprowadzonych eksperymentów. Dla pełniejszej automatyzacji stworzono jeszcze skrypt powłoki uruchamiający testy dla zbiorów testowych (`run_tests.sh`).

Pełny kod źródłowy, skrypty oraz pliki LaTeX sprawozdania znajdują się na repozytorium GitHub pod adresem:

<https://github.com/kplawik/IAD-podyplomowe>

# 3 Eksperymenty i wyniki

W celu przetestowania stworzonego rozwiązania przeprowadzono siedem eksperymentów dla różnych wariantów zaszumienia, różnych czcionek oraz zestawów kilku znaków. Przebieg i wyniki eksperymentów opisano w kolejnych podrozdziałach.

### 3.1 Test poprawności działania

Nim przystąpiłem do wykonywania eksperymentów dokonałem sprawdzenia poprawności działania sieci. W celu tym uruchomiłem skrypt podając dane treningowe również jako dane testowe. Zgodnie z przewidywaniami litery zostały rozpoznane poprawnie ze 100% pewnością:

Network test

```
letter A, noise level 0% -> letter A, noise level 0%, confidence: 1.000000  
letter B, noise level 0% -> letter B, noise level 0%, confidence: 1.000000  
letter C, noise level 0% -> letter C, noise level 0%, confidence: 1.000000
```

### 3.2 Eksperyment nr 1

Eksperyment nr 1 potraktowałem jako rozgrzewkę. Utworzyłem trzy znaki o różnym stopniu zaszumienia oraz dodałem dwa inne znaki - D oraz X.

#### 3.2.1 Założenia

Zakładam że litery zostaną rozpoznane poprawnie ale poziom pewności będzie proporcjonalny do stopnia zaszumienia. Spodziewam się również że litera D zostanie rozpoznana jako B ze względu na podobieństwo kształtu. Litera X zostanie rozpoznana jako któraś z testowych ale zapewne z niskim stopniem pewności.

#### 3.2.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

Eksperyment 1

```
letter A, noise level 15% -> letter A, noise level 0%, confidence: 0.916784  
letter B, noise level 25% -> letter B, noise level 0%, confidence: 0.851735  
letter C, noise level 50% -> letter C, noise level 0%, confidence: 0.695136  
letter D, noise level 0% -> letter B, noise level 0%, confidence: 0.981603  
letter X, noise level 0% -> letter A, noise level 0%, confidence: 0.969461
```

#### 3.2.3 Rezultat

Zgodnie z przewidywaniami litery A, B i C zostały rozpoznane poprawnie a wraz ze wzrostem stopnia zaszumienia stopień pewności malał. Litera D została rozpoznana jako B w dużym stopniu pewności (zapewne ze względu na podobieństwo wyglądu) a litera X jako A (zapewne ze względu na długie

skośne krawędzie) ale stopień pewności był nieznacznie niższy niż dla litery D.

### 3.3 Eksperyment nr 2

Zgodnie z treścią zadanie ekperyment wykorzystał zaszumienie 10%. Literę D roztawiłem ale X usunąłem ze względu na niski stopień pewności.

#### 3.3.1 Założenia

Spodziewam się poprawnego rozpoznania liter A, B i C i stopnia pewności powyżej 90%.



Rysunek 1: Litera A ze stopniem zaszumienia 10%.

#### 3.3.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

Eksperyment 2

```
letter A, noise level 10% -> letter A, noise level 0%, confidence: 0.947077
letter B, noise level 10% -> letter B, noise level 0%, confidence: 0.945833
letter C, noise level 10% -> letter C, noise level 0%, confidence: 0.947273
letter D, noise level 10% -> letter B, noise level 0%, confidence: 0.930001
```

#### 3.3.3 Rezultat

Zgodnie z przewidywaniami llitery A,B i C roztały rozpoznane poprawnie a stopień pewności oscylował w granicach 94%. Litera D tak jak ostatnio - stopień pewności niższy od B. Analiza wzrokowa obrazów testowych wykazała lekkie zaszumienie a litery czytelne bez majmniejszego problemu.

## 3.4 Eksperyment nr 3

Zgodnie z treścią zadanie ekperyment wykorzystał zaszumienie 30%.

### 3.4.1 Założenia

Litery A, B i C zostaną rozpoznane poprawnie z dokładnością ok. 75%. Litera D minimalnie niżej.



Rysunek 2: Litera A ze stopniem zaszumienia 30%.

### 3.4.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

Eksperyment 3

```
letter A, noise level 30% -> letter A, noise level 0%, confidence: 0.825020
letter B, noise level 30% -> letter B, noise level 0%, confidence: 0.820656
letter C, noise level 30% -> letter C, noise level 0%, confidence: 0.825726
letter D, noise level 30% -> letter C, noise level 0%, confidence: 0.810182
```

### 3.4.3 Rezultat

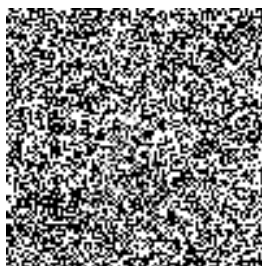
Litery A, B i C rozpoznane poprawnie z dokładnością nieco wyższą niż się spodziewałem (ta oscylowała w ok. 82%). Litera D rozpoznana jako C z dokładnością 81% czyli pozostawienie jej w kolejnych iteracja jest już tylko ciekawostką. Wygenerowane pliki graficzne czytelne dla ludzkiego oka choć szum wyraźnie je utrudniał.

## 3.5 Eksperyment nr 4

Ekperyment wykorzystał zaszumienie 50%.

### 3.5.1 Założenia

Zakładam poziom rozpoznania w granicach 50%.



Rysunek 3: Litera A ze stopniem zaszumienia 50%.

### 3.5.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

Eksperyment 4

```
letter A, noise level 50% -> letter A, noise level 0%, confidence: 0.693884
letter B, noise level 50% -> letter A, noise level 0%, confidence: 0.691056
letter C, noise level 50% -> letter C, noise level 0%, confidence: 0.695136
letter D, noise level 50% -> letter C, noise level 0%, confidence: 0.691716
```

### 3.5.3 Rezultat

Podany poziom wynosił 69%. z tym że litera B nie została rozpoznana poprawnie (jako A). Litera D jako C (czyli jak w poprzednim przypadku). Wygenerowane pliki graficzne praktycznie nieczytelne dla ludzkiego oka.

## 3.6 Eksperyment nr 5

Ekperyment wykorzystał zaszumienie 90%.

### 3.6.1 Założenia

Spodziewam się niepoprawności z rozpoznaniu i niskich wskazań pewności (pewnie ok. 50%).



Rysunek 4: Litera A ze stopniem zaszumienia 90%.

### 3.6.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

Eksperyment 5

```
letter A, noise level 90% -> letter C, noise level 0%, confidence: 0.345270
letter B, noise level 90% -> letter A, noise level 0%, confidence: 0.335060
letter C, noise level 90% -> letter A, noise level 0%, confidence: 0.340770
letter D, noise level 90% -> letter A, noise level 0%, confidence: 0.353479
```

### 3.6.3 Rezultat

Żadna z liter nie rozpoznana poprawnie. Podane współczynniki pewności ok 35%. Naaoczne oględziny wygenerowanych plików pokazują że są już one prawie negatywami liter treningowych.

## 3.7 Eksperyment nr 6

Eksperyment dodatkowy sprawdzający jak sieć poradzi sobie z zestawami liter - jak np. Ai albo Bo.

### 3.7.1 Założenia

Ze względu na brak zmienionej centryzacji spodziewam się dość dużej rozpoznawalności bo dodatkowe litery będą umieszczone obok zadanych (tożsaczym z treningowymi).

### 3.7.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:



Rysunek 5: Litery A oraz i.

#### **Eksperyment 6**

```
letter Ai, noise level 0% -> letter A, noise level 0%, confidence: 0.989587  
letter Bo, noise level 0% -> letter B, noise level 0%, confidence: 0.982996  
letter Cy, noise level 0% -> letter C, noise level 0%, confidence: 0.983019  
letter DT, noise level 0% -> letter B, noise level 0%, confidence: 0.961837
```

#### **3.7.3 Rezultat**

Zgodnie z założeniami - dodatkowe litery stały obok wzorcowych dzięki czemu sieć prawidłowo rozpoznała litery dominujące (A, B, C) a dodatkowe (i, o, y, T) stanowiły jedynie "szum" dla stopnia pewności, który wynosił 96-98%.

### **3.8 Eksperyment nr 7**

Tym razem dane testowe wygenerowane zostały z użyciem czcionki consolas.ttf (poprzednie times.ttf).

#### **3.8.1 Założenia**

Spodziewam się błędów i dużego spadku pewności rozpoznania.



Rysunek 6: Litera A: po lewej treningowa, po prawe testowa pisana inną czcionką



### 3.8.2 Przebieg

Po wygenerowaniu zadanych danych testowych i uruchomieniu testów otrzymaliśmy następujący listing:

`Eksperyment 7`

```
letter A, noise level 0% -> letter A, noise level 0%, confidence: 0.967252
letter B, noise level 0% -> letter C, noise level 0%, confidence: 0.960595
letter C, noise level 0% -> letter C, noise level 0%, confidence: 0.968028
```

### 3.8.3 Rezultat

Dwie litery rozpoznane poprawnie - A i C. B błędnie rozpoznana jako C z zaskakująco wysokim stopniem dokładności - 96%.

## 4 Wnioski

Wnioski z przeprowadzonych eksperymentów dowodzą, że:

1. Wraz ze wzrostem zaszumienia maleje zarówno prawidłowość klasyfikacji jak i stopień podobieństwa (confidence)
2. Wysoki stopień podobieństwa (confidence) nie musi być gwarancją poprawnej klasyfikacji testowanego znaku
3. 30% zaszumienia wydaje się być umowną granicą dla której sieć MADALINE jest w stanie poprawnie rozpoznawać znaki. Mowa tu oczywiście na analizowanym przypadku, ponieważ dla innych czcionek, znaków czy metod wprowadzania szumu granica ta może być inna.
4. Przy niewielkim zaszumieniu klasyfikacja przebiega poprawnie z wysokim stopniem podobieństwa.