**Maciej Oziębły**
**Nr indeksu: 184147**

# Programowanie sieciowe
# Sprawozdanie 3

## *Zakres sprawozdania:*

- **redukcja wymiarowości danych wejściowych (metoda PCA)**
- **rozpoznawanie liter za pomocą sieci RBF**

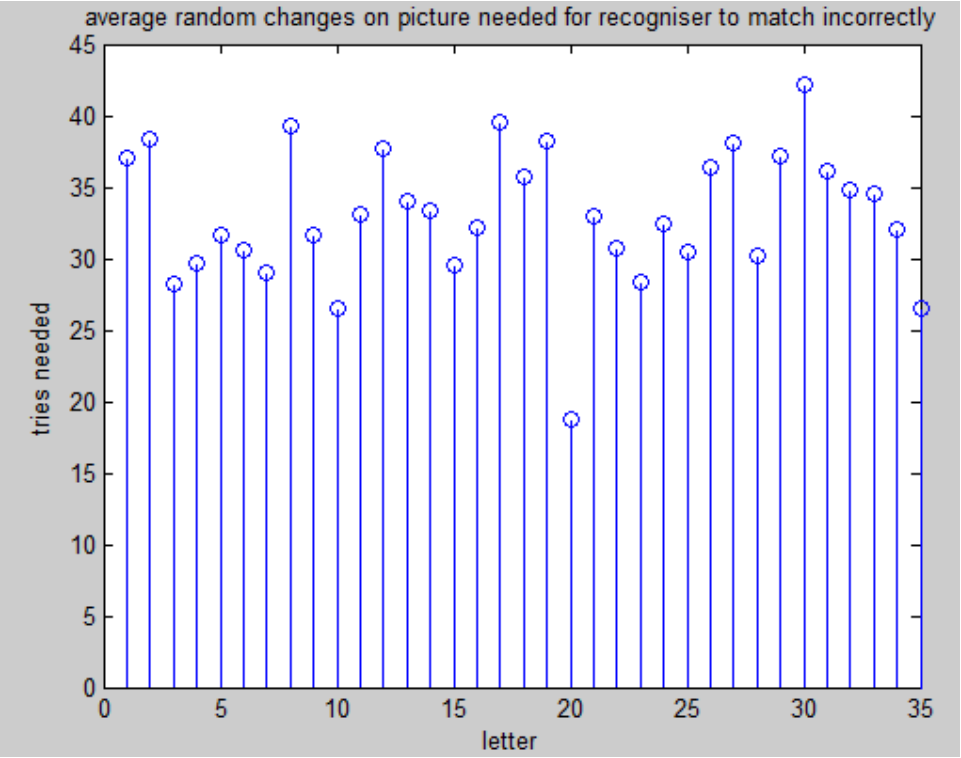
## Testy numeryczne:

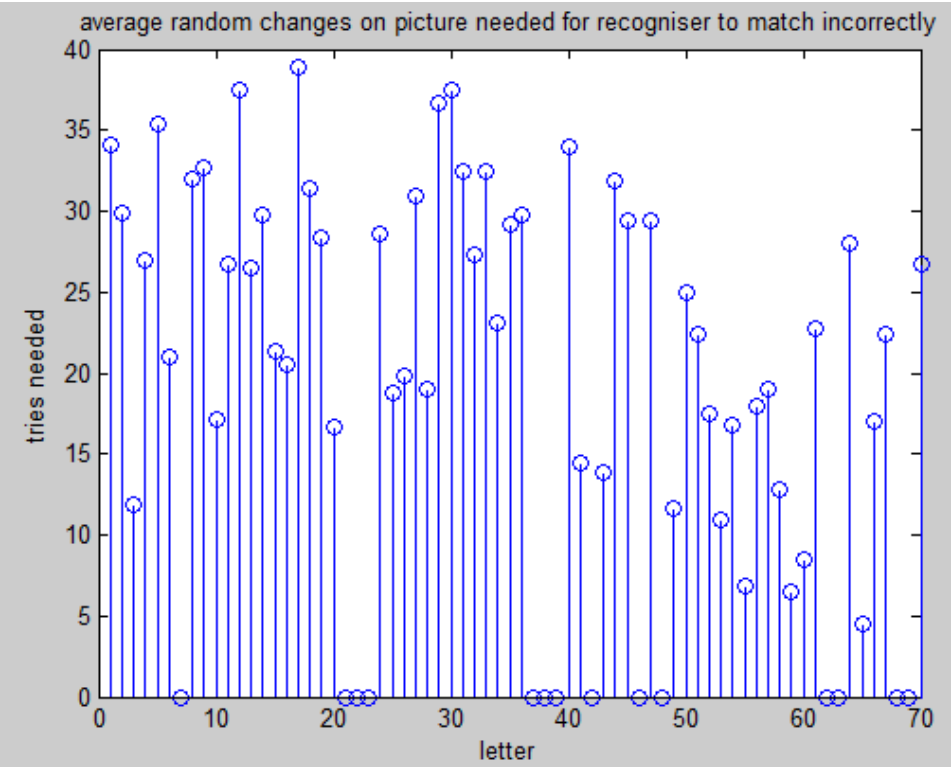
W celu przetestowania działania algorytmów wykorzystałem skrypt „testerka.m”, który był przeze mnie wykorzystywany na obu poprzednich ćwiczeniach (lekko zmodyfikowany):

```
letters = load_letters_definitions();


tries_per_letter = 50;
certainity_level = 0.80;
avg_tries_per_letter = zeros(size(letters,2),1);

[cut_letters_definitions, A_matrix] = cut_dimensions(load_letters_definitions(),
certainity_level);
weights = calc_weights_matrix(cut_letters_definitions);
for letter_no = 1:size(letters,2)
    tries = 0;
    letter_no
    for i = 1:tries_per_letter
        letter = letters(:, letter_no);
        random_changes_before_fail = 0;
        match_failed = 0;
        changed_indexes = zeros(100,1);
        while(match_failed == 0 && random_changes_before_fail ~= 100)
            if(association_recogniser_optimized(letter, weights, A_matrix) ~=
mod((letter_no-1), 35)+1)
            %if(association_recogniser(letter) ~= mod((letter_no-1), 35)+1)
                match_failed = 1;
            else
                % generate random index not changed yet
                rand_index = randi(100);
                while(max(ismember(changed_indexes, rand_index)) ~= 0)
                    rand_index = randi(100);
                end
                changed_indexes(length(changed_indexes)+1) = rand_index;

                letter(rand_index) = letter(rand_index) * -1;
                random_changes_before_fail = random_changes_before_fail + 1;
            end
        end
        tries = tries + random_changes_before_fail;
    end
    avg_tries_per_letter(letter_no) = tries / tries_per_letter;
end

stem(avg_tries_per_letter)
title('average random changes on picture needed for recogniser to match
incorrectly')
xlabel('letter')
ylabel('tries needed')
```

## Wyniki testów (redukcja wymiarów)

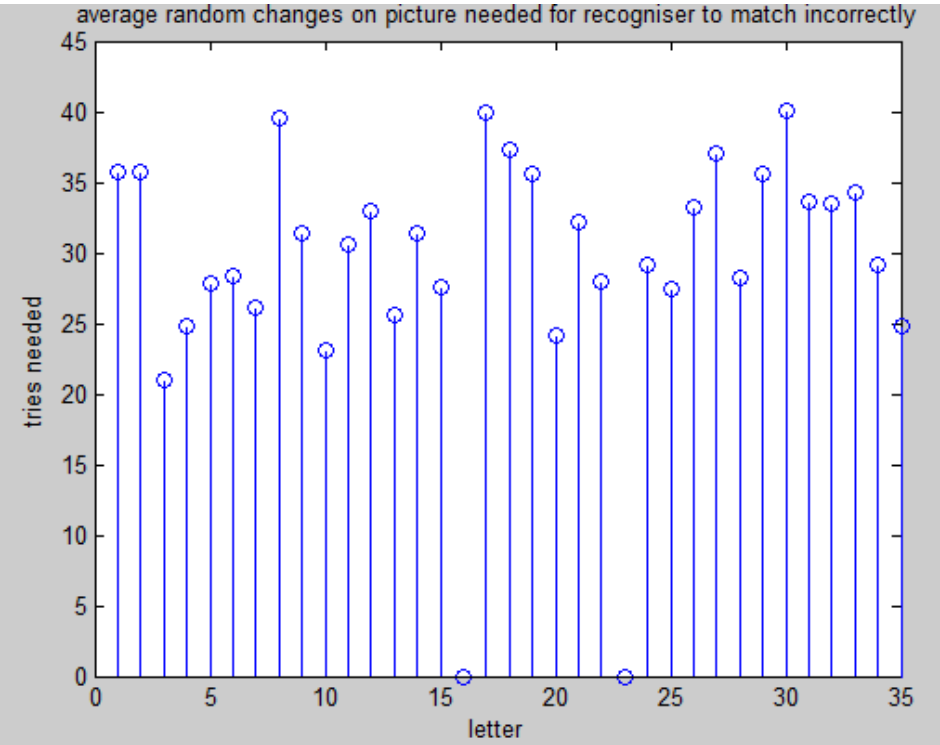
Dla stopnia pewności wynoszącego 1 (oczekiwane wyniki takie same jak przed zmniejszaniem wymiarowości):

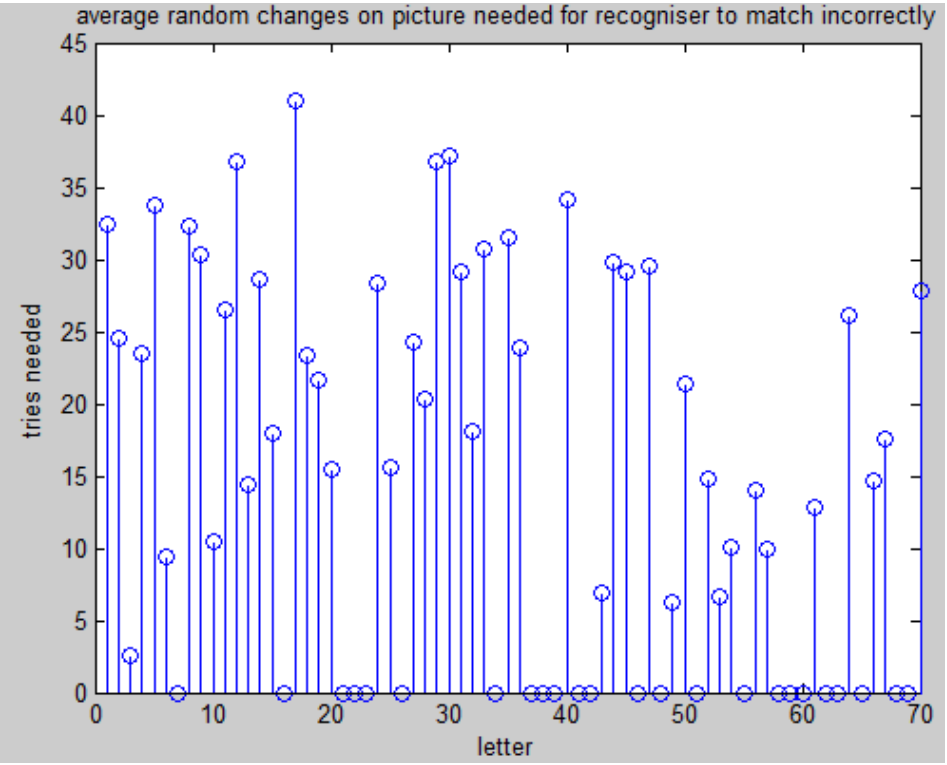



Wyniki spełniły moje oczekiwania – sieć radzi sobie z rozpoznawaniem tak samo jak przed redukcją wymiarów danych wejściowych.

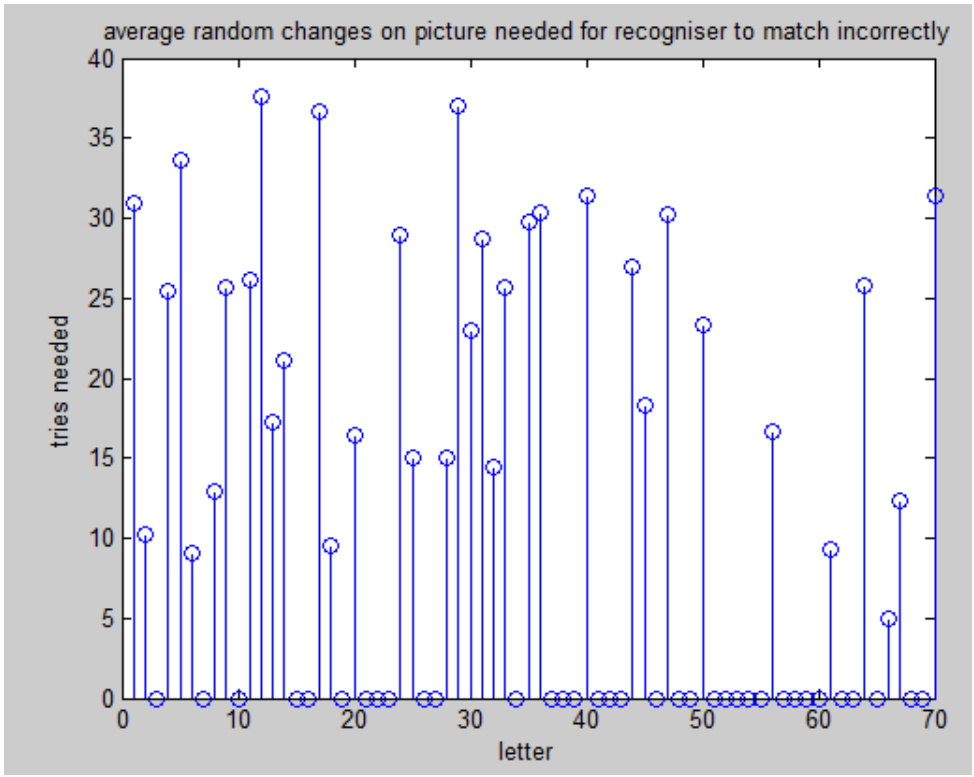
Po zmniejszeniu stopnia pewności do 0.95:



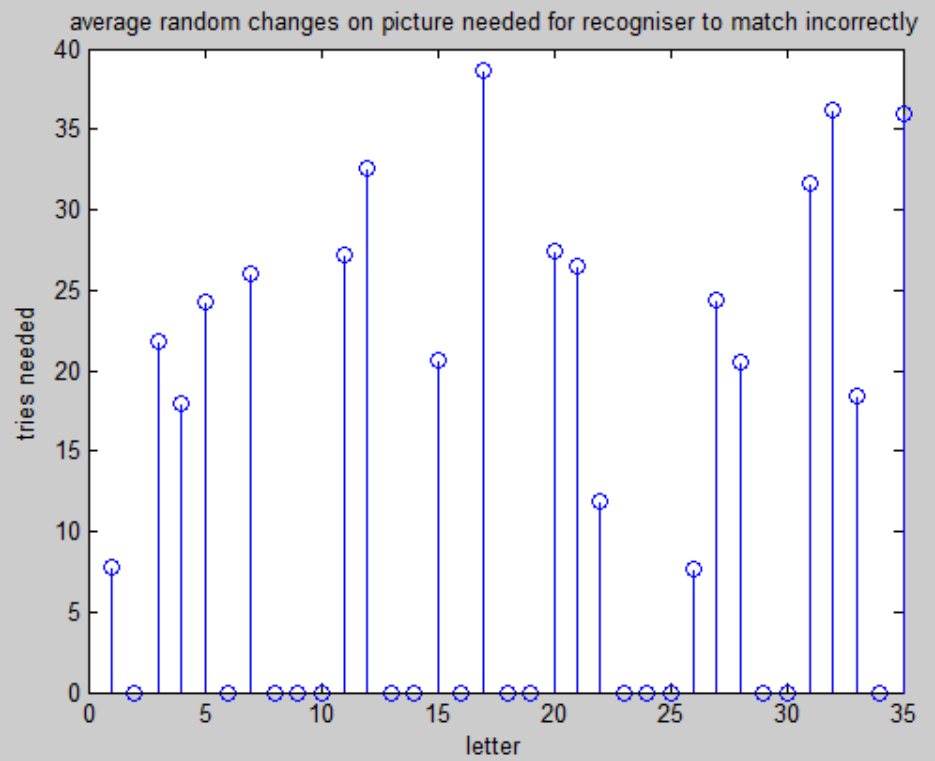

Przy obecności większej liczy wzorców widoczne jest znaczne pogorszenie wyników rozpoznawania liter. Dla małej ilości liter sieć nadal dobrze radzi sobie z rozpoznawaniem liter, jednak w 2 przypadkach od razu pojawiają się błędy.

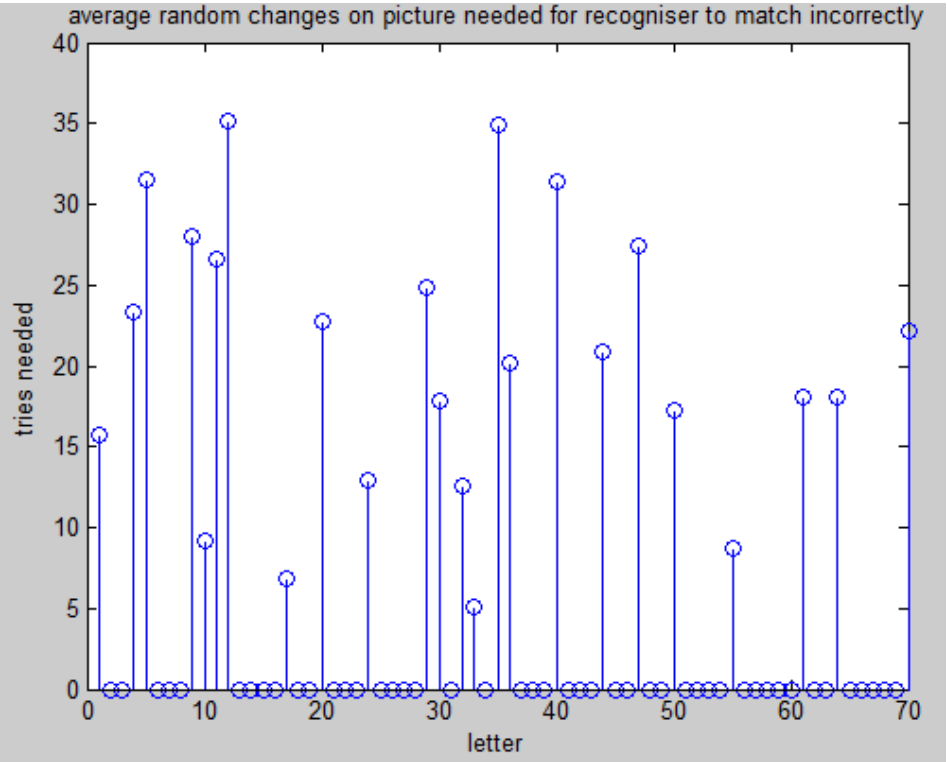
Dla poziomu pewności 0.9:


Sieć z dużą liczbą wzorców praktycznie przestała działać – dla małej liczby wzorców nie widać zmian.

Dla stopnia pewnośći 0.8:

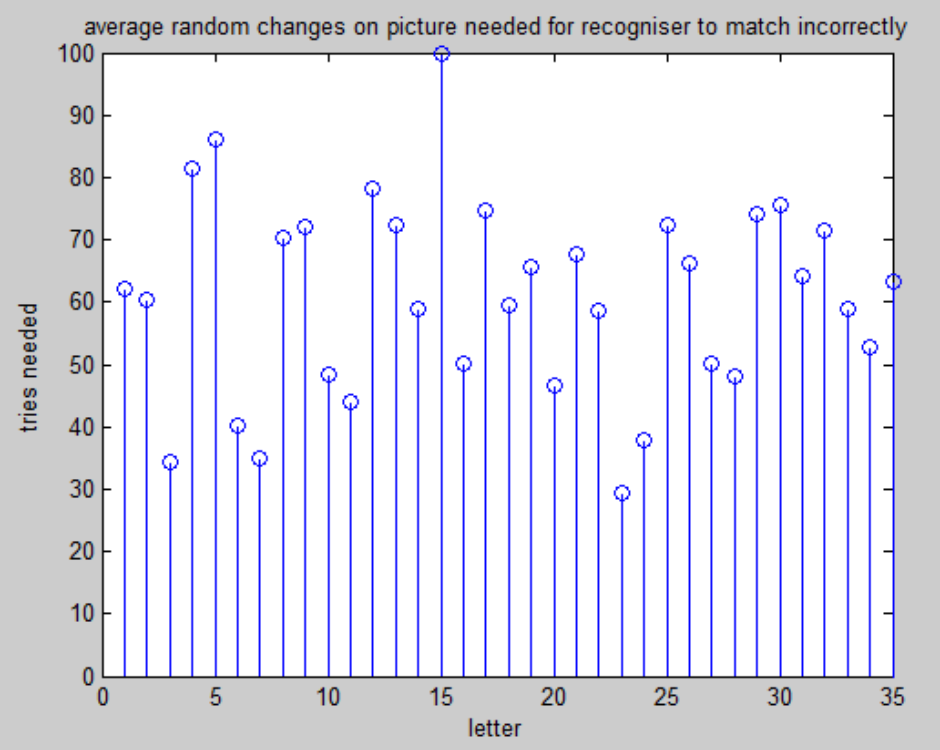



Przy stopniu pewnościu wynoszącym 0.8 sieć przestała radzić sobie z rozpoznawaniem liter zarówno dla małej, jak i dla dużej liczby wzorców.

## Sieć RBF – wyniki testów:

Wykorzystałem sieć RBF do rozpoznawania wielkich liter (tak jak w poprzednim zadaniu):

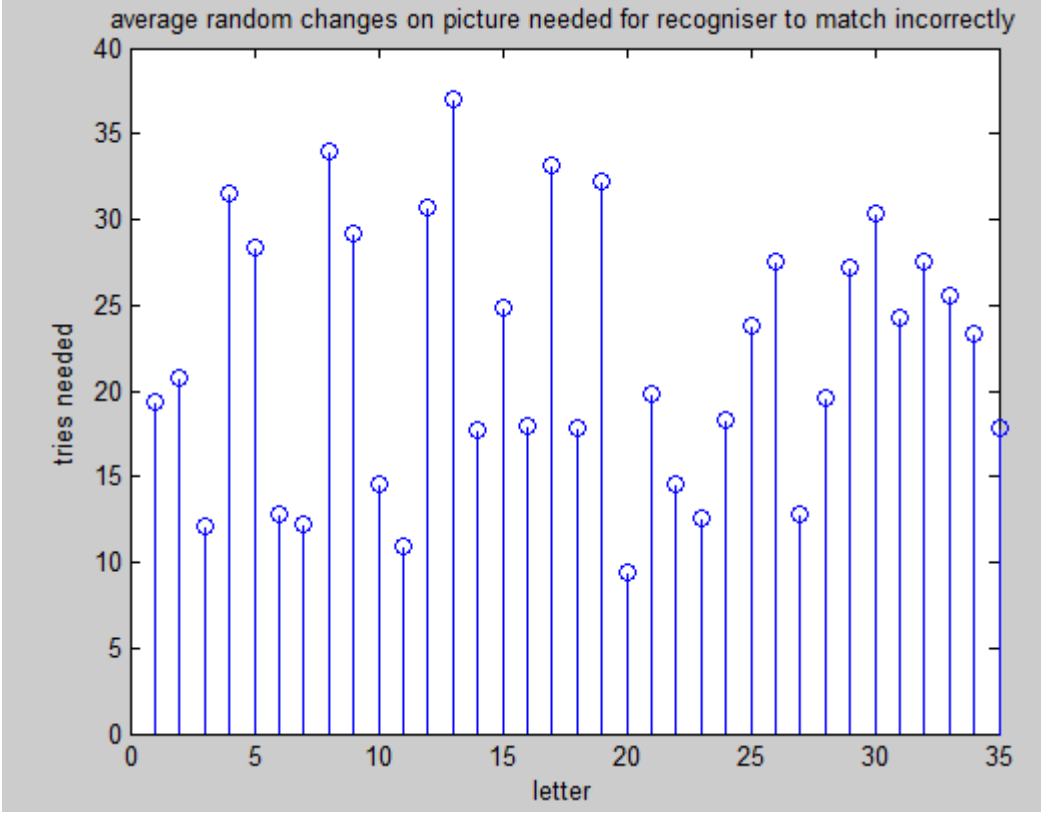



Okazuje się, że zastosowanie sieci RBF daje znacznie lepsze wyniki niż testowane wcześniej sieci asocjacyjne.

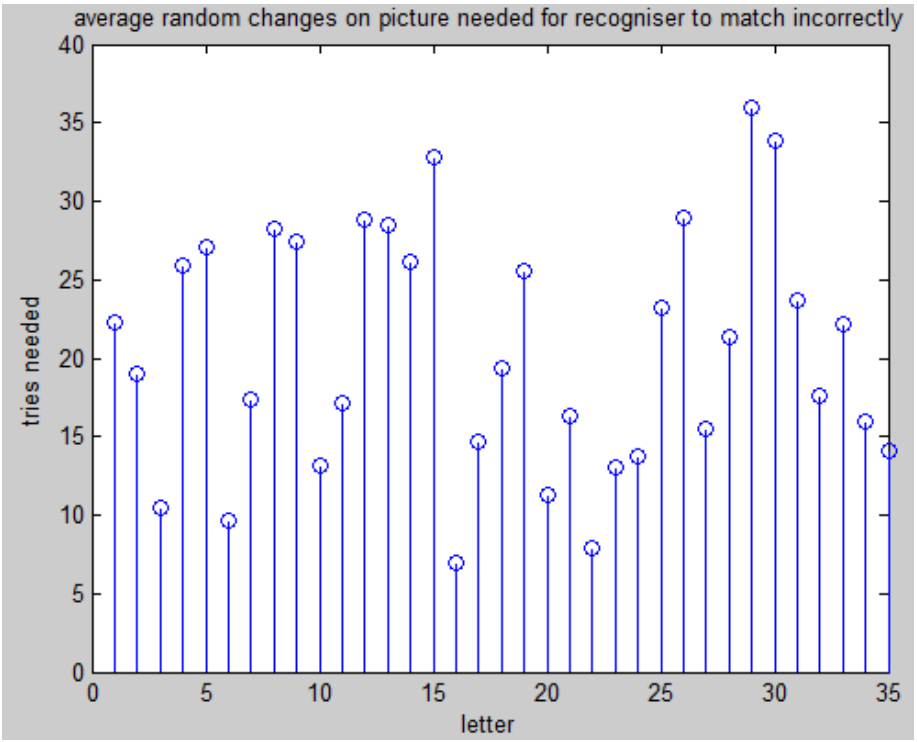
Po ogarniczeniu wymiarowości za pomocą metody PCA (przy stopniu pewności 0.95):


Widać tutaj znaczne pogorszenie wyników, ale algorytm zaczął działać znacznie szybciej.

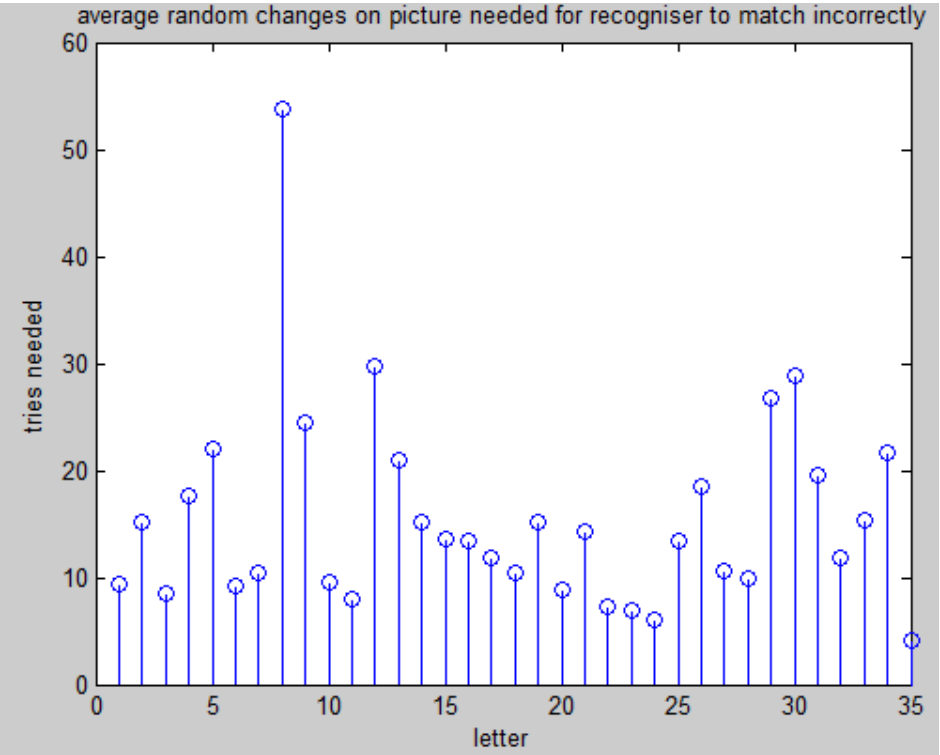
Po zmniejszeniu stopnia pewności do 0.9:




Po zmniejszeniu stopnia pewności do 0.85:



Po zmniejszeniu stopnia pewności do:




## Wnioski:

- ograniczenie liczby wymiarów danych wejściowych do sieci neuronowej metodą PCA pozwala na znaczne przyspieszenie działania algorytmu.
- korzystając z metody PCA optymalnie zmniejszamy liczbę danych (dla zadanego stopnoia pewności), ponieważ w tej metodzie maksymalizujemy wariancję danych po przekształceniu, co zapewnia zachowanie maksymalnie dużej liczby informacji.
- po zmniejszeniu stopnia pewności poniżej pewnego stopnia (w moim przypadku dla samych wielkich liter poniżej 0.9, dla małych i wielkich liter poniżej 0.95) sieć przstaje radzić sobie z rozpoznawaniem liter. Po nieznacznym zmniejszeniu stopnia pewności uzyskujemy duże przyspieszenie działania algorytmu, a działa on tylko nieznacznie gorzej.
- zastosowanie sieci RBF daje lepsze rezultaty niż wykorzystywane wcześniej liniowe i asocjacyjne sieci przy rozpoznawaniu liter
- skuteczność sieci RBF zmniejsza się bardzo wolno wraz ze zmniejszaniem wymiarowości danych wejściowych metodą PCA – nawet po zmniejszeniu stopnia pewności do 0.75 sieć RBF radziła sobie z rozpoznawaniem wszystkich liter do do pewneg – całkiem sporego stopnia zaszumienia.