

1 Opis

Na wykresach w sekcji nr. 2 widać, że występuje zależność pomiędzy skutecznością danej konfiguracji na zbiorze walidacyjnym i testowym.

Można na tej informacji oprzeć metodę adaptacyjną, bo zbiorem walidacyjnym dysponujemy w trakcie trenowania metody. I w większości przypadków byłoby to skuteczne podejście.

Niestety na wykresach w sekcji nr. 3, w szczególności na wykresie VGG_CIFAR100vsMNIST pokazana jest pułapka tego podejścia.

Czasami słabe w ogólności konfiguracje pozwalają znaleźć najlepszy próg oddzielający zbiór treningowy od zupełnie różnego zbioru walidacyjnego takiego jak NormalNoise.

Pomysły mam dwa

1. Algorytm treningu i działania metody byłby następujący:

- Dla każdej konfiguracji z hipersiatki możliwych konfiguracji wyznacz threshold i skuteczność vs dataset walidacyjny
- Dla n najlepszych warstw wybierz najlepszą konfigurację dla danej warstwy i zapamiętaj n zbiorów patternów treningowych (jeden dla każdej konfiguracji) i threshold dla niej wyznaczony

Faza testowa:

- Wyznacz n patternów dla próbki testowej (jeden dla każdej konfiguracji; wszystkie n podczas jednego forward passu przez sieć rzecz jasna)
- Wyznacz dystans n patternów z odpowiadającymi zbiorami patternów znanych i porównaj z thresholdami
- Głosowanie warstw - jeśli i-ty pattern porównany z i-tym zbiorem znanych patternów jest poniżej thresholdu to warstwa głosuje, że pattern jest znany

Liczba n - branych pod uwagę warstw powinna być nieparzysta, albo trzeba by zastosować ważenie głosów (lepsza skuteczność na zbiorze walidacyjnym -> większa waga)

Moim zdaniem ma to potencjał dać dobre wyniki - dla takich przypadków jak CIFAR100vsNoise/MNIST da na pewno gorszy wynik, niż najlepsza ogólnie pojedyncza warstwa, lecz nie mamy dostępu do wyroczeni.

A dla pozostałych przypadków niewykluczone, że otrzymamy nawet lepsze wyniki dzięki fazie głosowania

2. By zniwelować niekorzystny efekt opisany wyżej przy uczeniu metody przeciw takim datasetom jak Normal-Noise można spróbować podczas treningu wykorzystać substytut podobnego datasetu w postaci próbek należących do części klas znanego datasetu.

Wyglądałoby to mniej więcej tak:

- wybieramy n klas zbioru treningowego np. 30%
- szukamy najlepszej konfiguracji podobnie jak wcześniej, tylko wybór opieramy jeszcze na informacji, która warstwa najlepiej separowała znany zbiór od sztucznie wygenerowanego podobnego nieznanego zbioru
- założymy, że zbiorem treningowym jest CIFAR100
- wybieramy klasy 80-100 jako nieznanne i tworzymy z nich zbiór walidacyjny; zapamiętujemy patterny pozostałych 80 klas
- gdy sieć sklasyfikuje jakąś próbkę jako jedną z klas 80-100, losowo wybieramy, od której klasy zapamiętanych patternów (0-79) liczymy odległość Hamminga

Nie jestem w stanie przewidzieć, czy ten sposób by coś nam dał, ale może udałoby się dzięki temu wybierać lepsze konfiguracje.

2 Wykresy konfiguracji zagregowane po wszystkich datasetach walidacyjnych (20 najlepszych wyników)

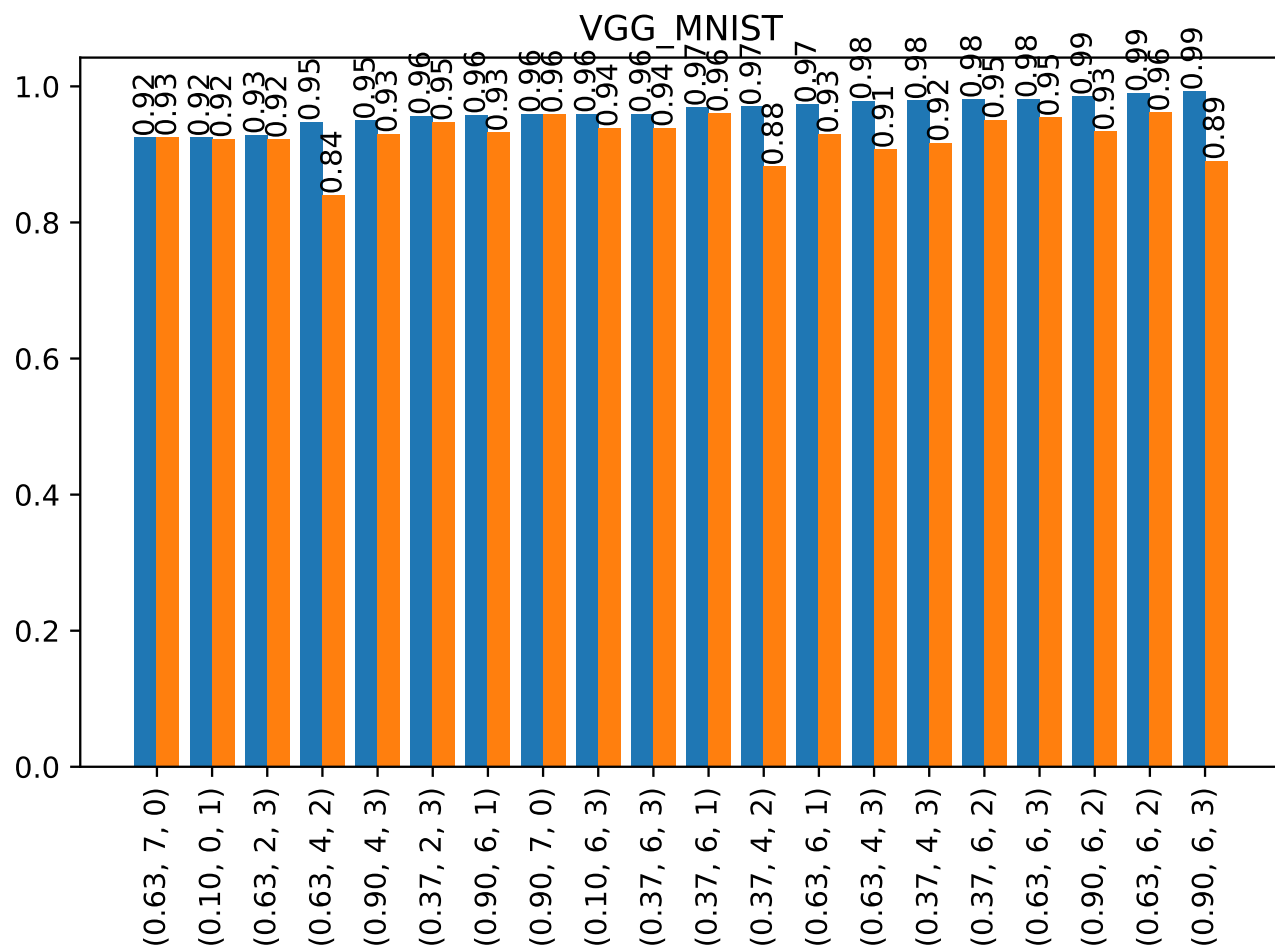


Figure 1: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

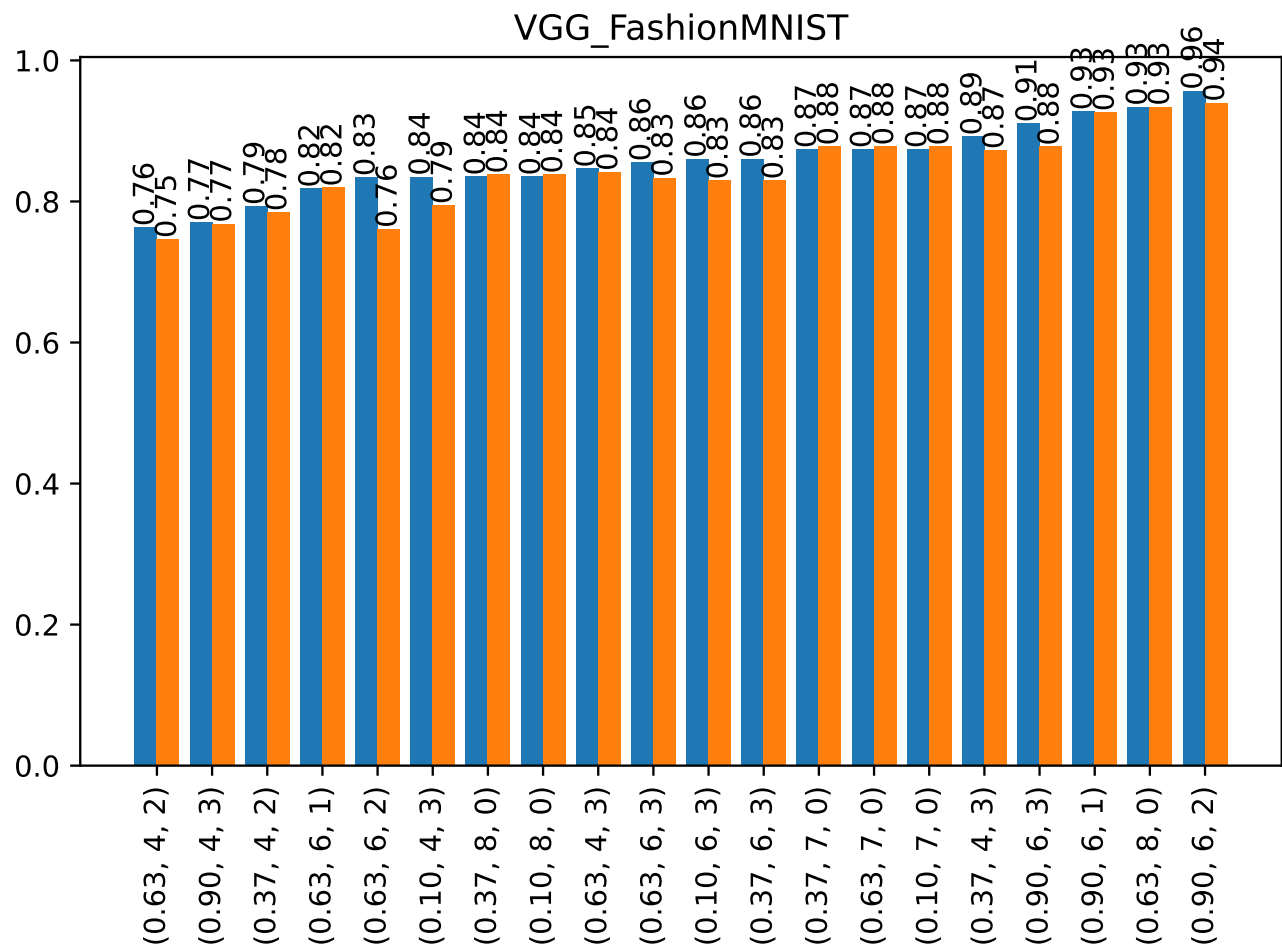


Figure 2: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

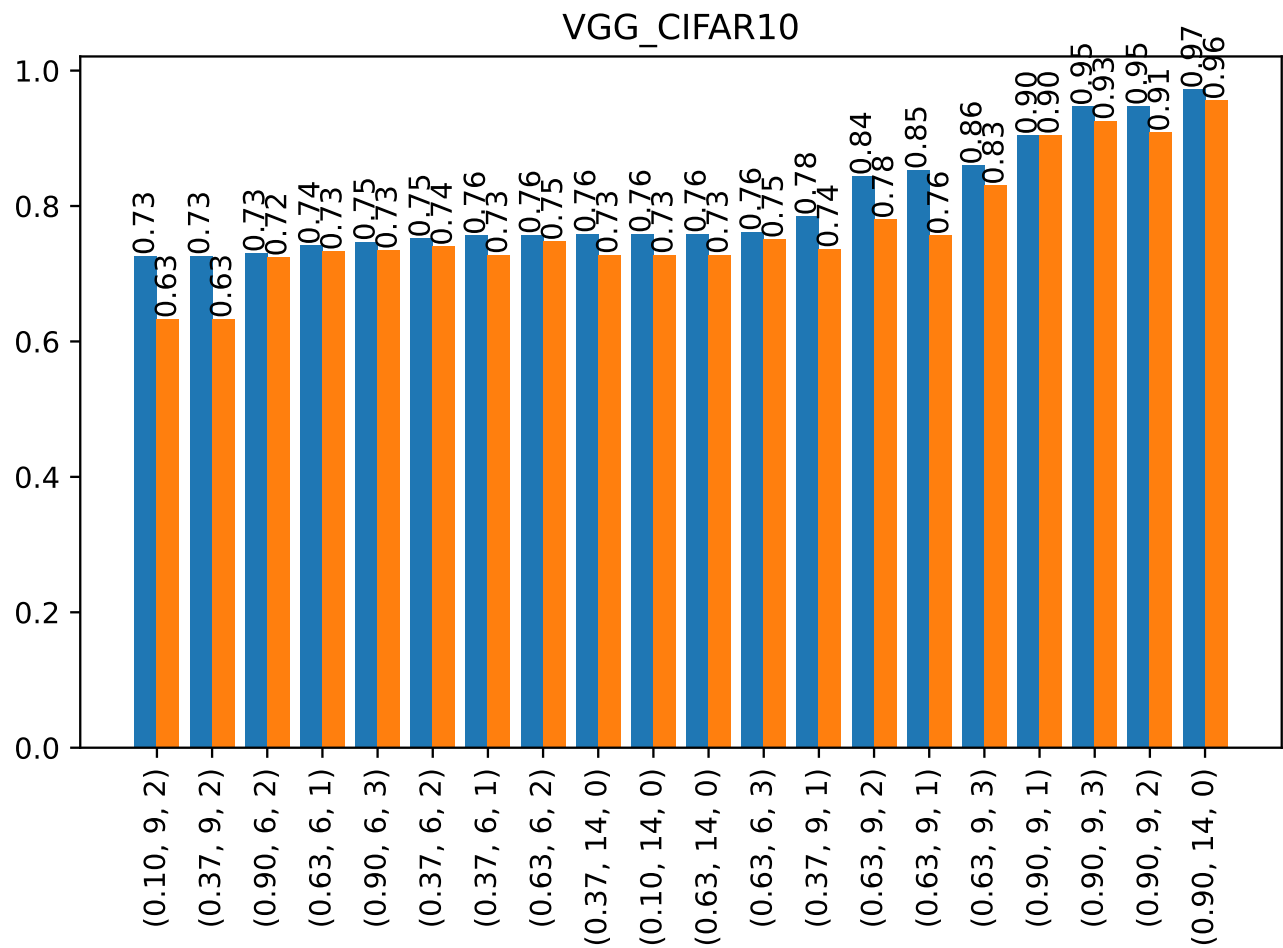


Figure 3: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

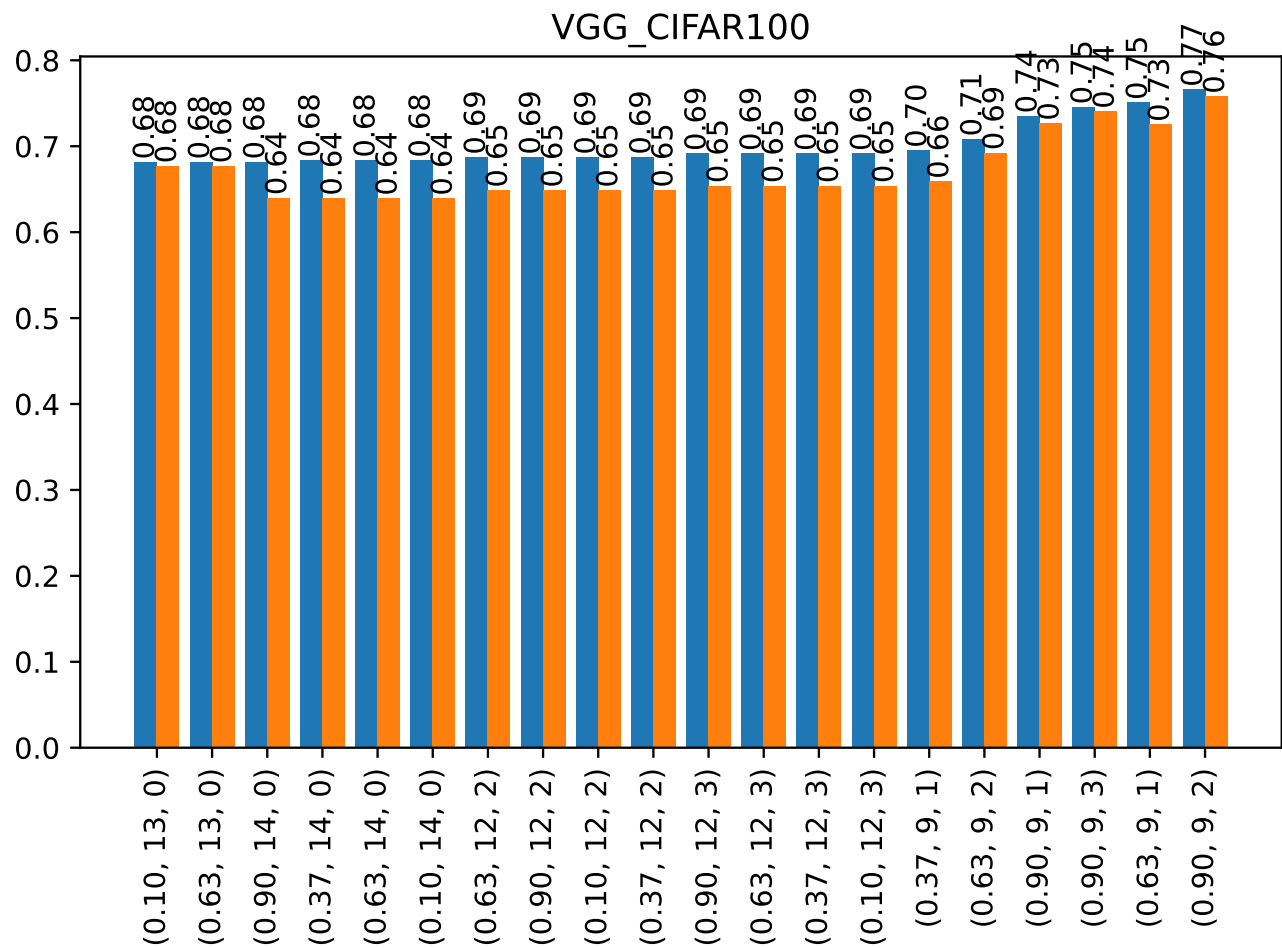


Figure 4: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

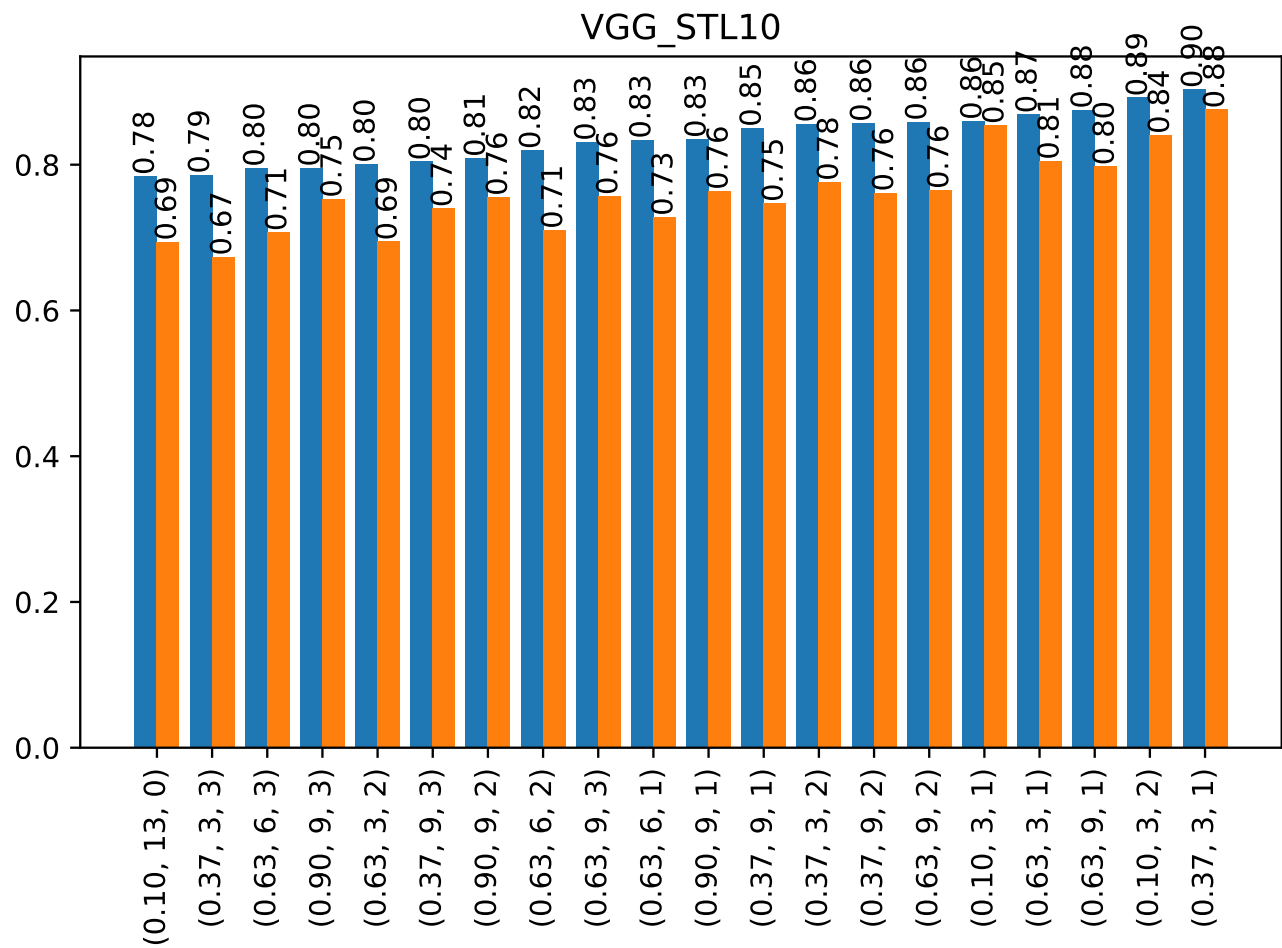


Figure 5: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

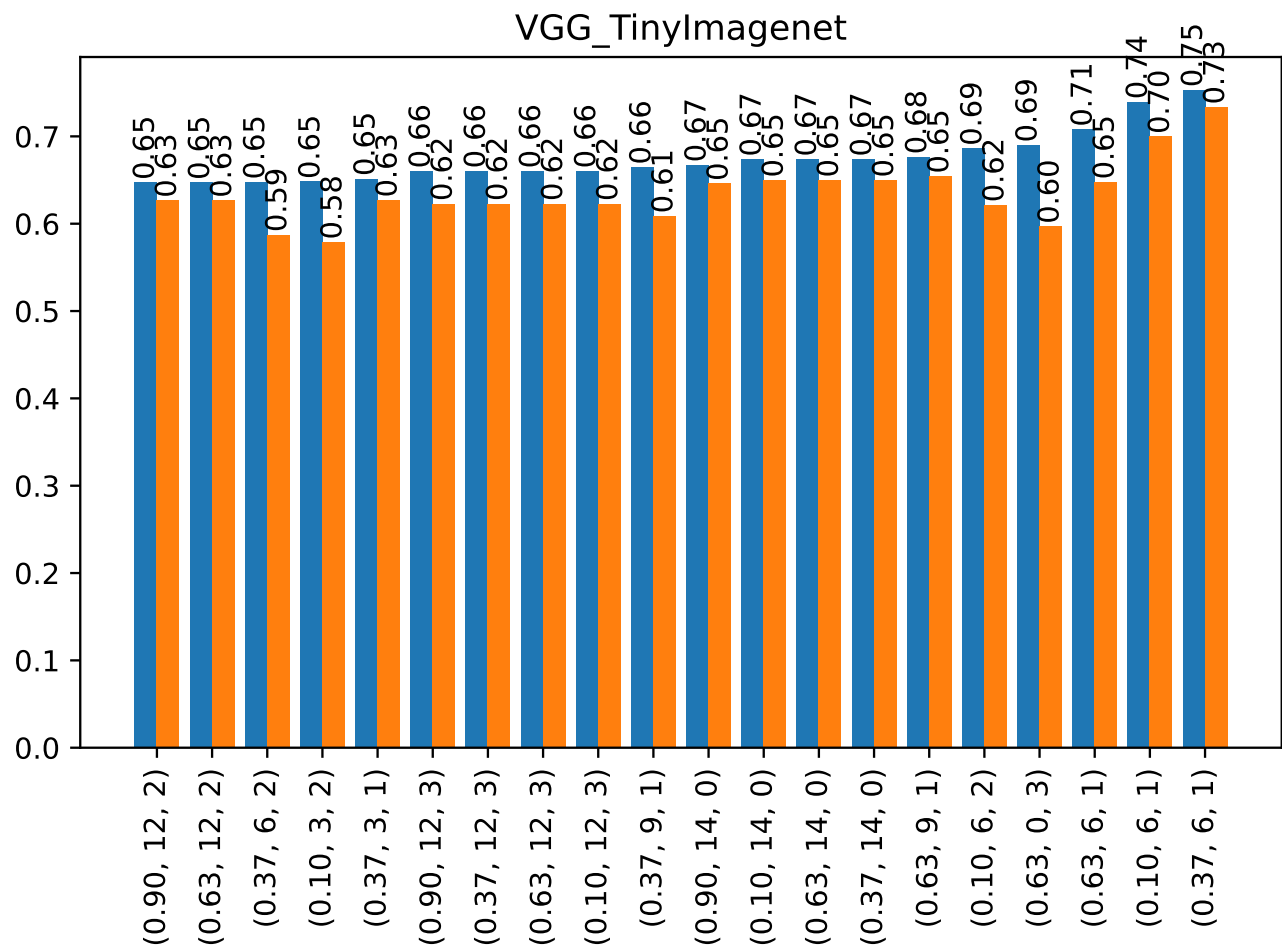


Figure 6: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

3 Wykresy konfiguracji (20 najlepszych wyników)

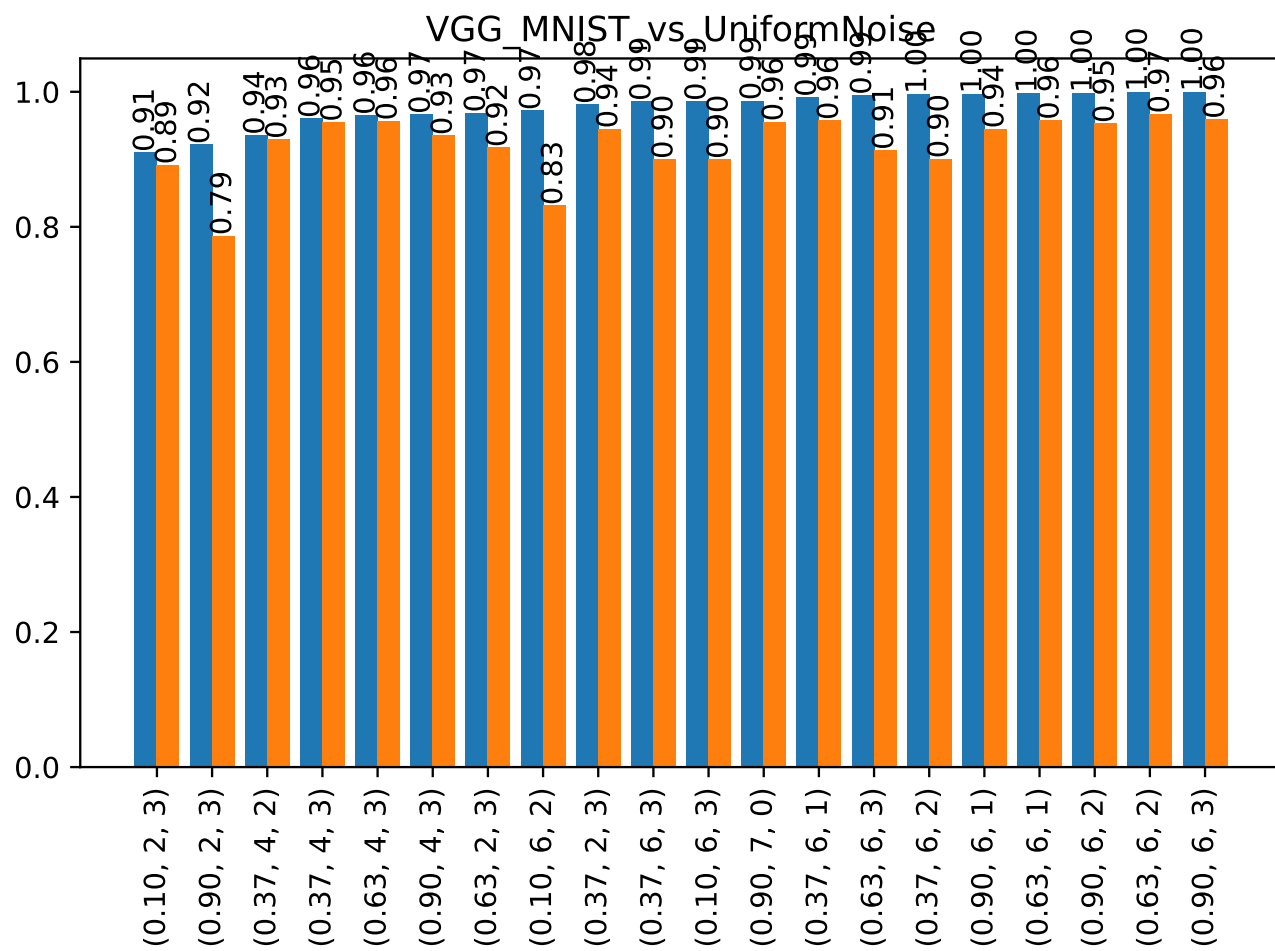


Figure 7: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

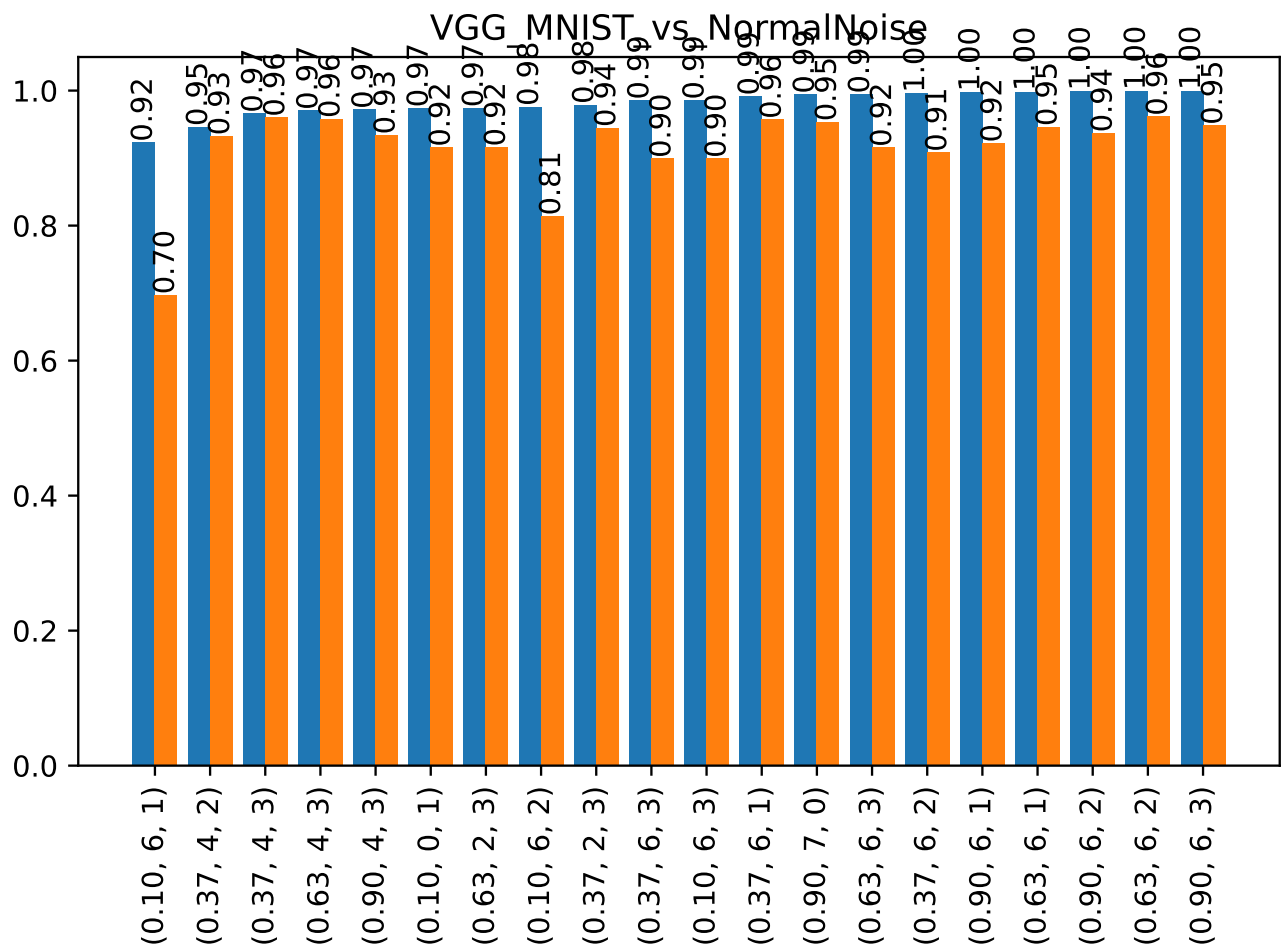


Figure 8: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

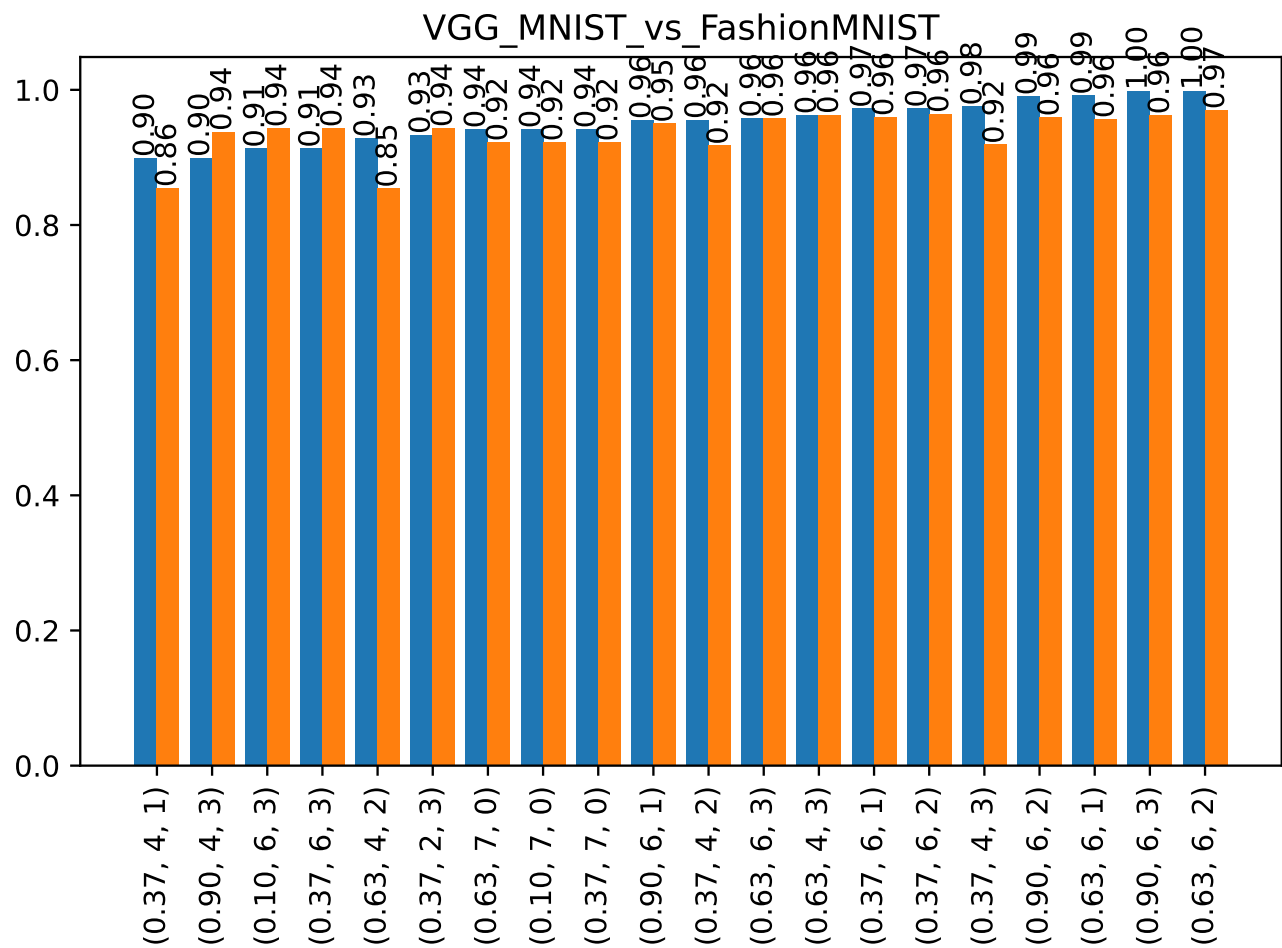


Figure 9: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

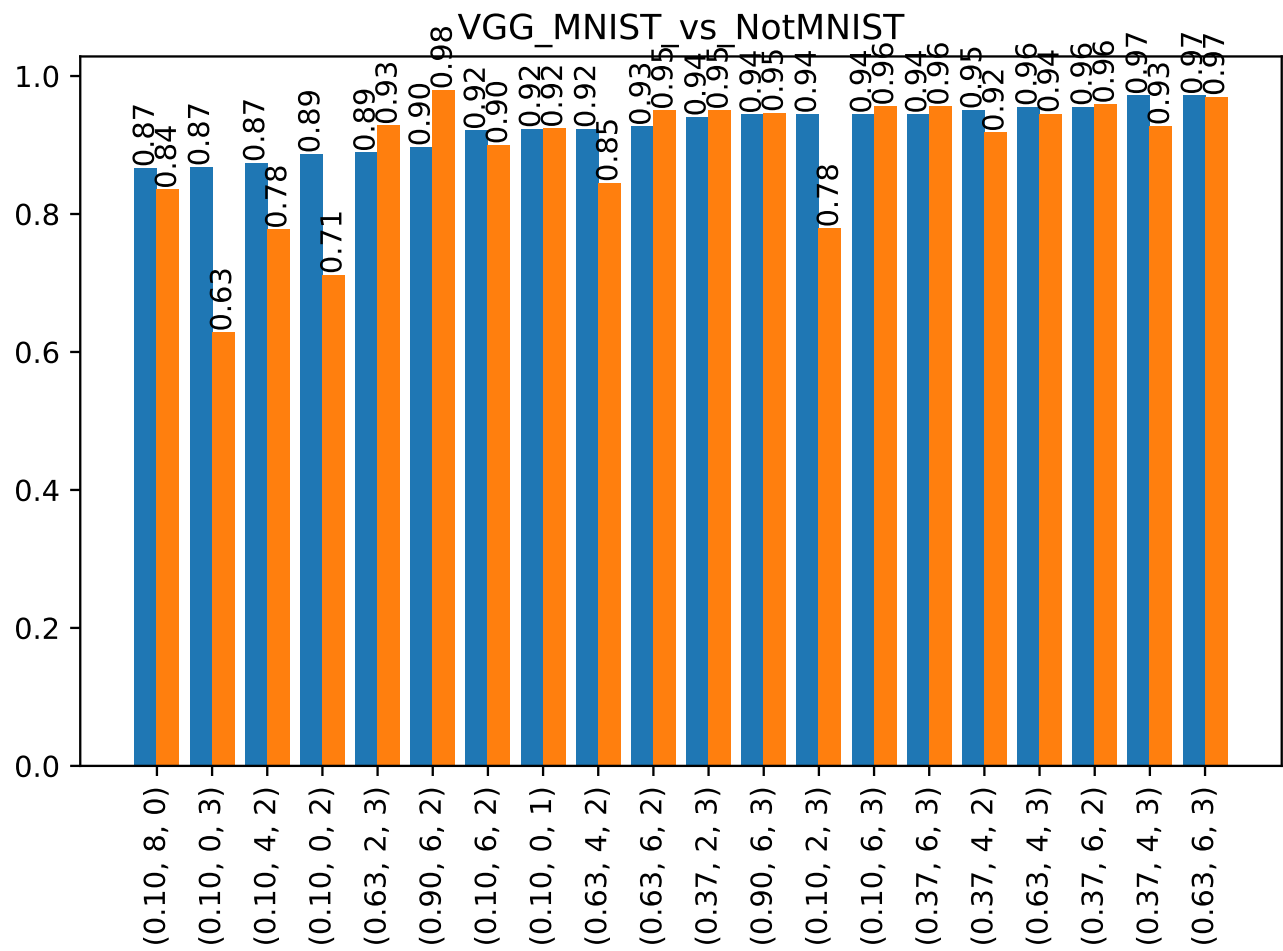


Figure 10: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

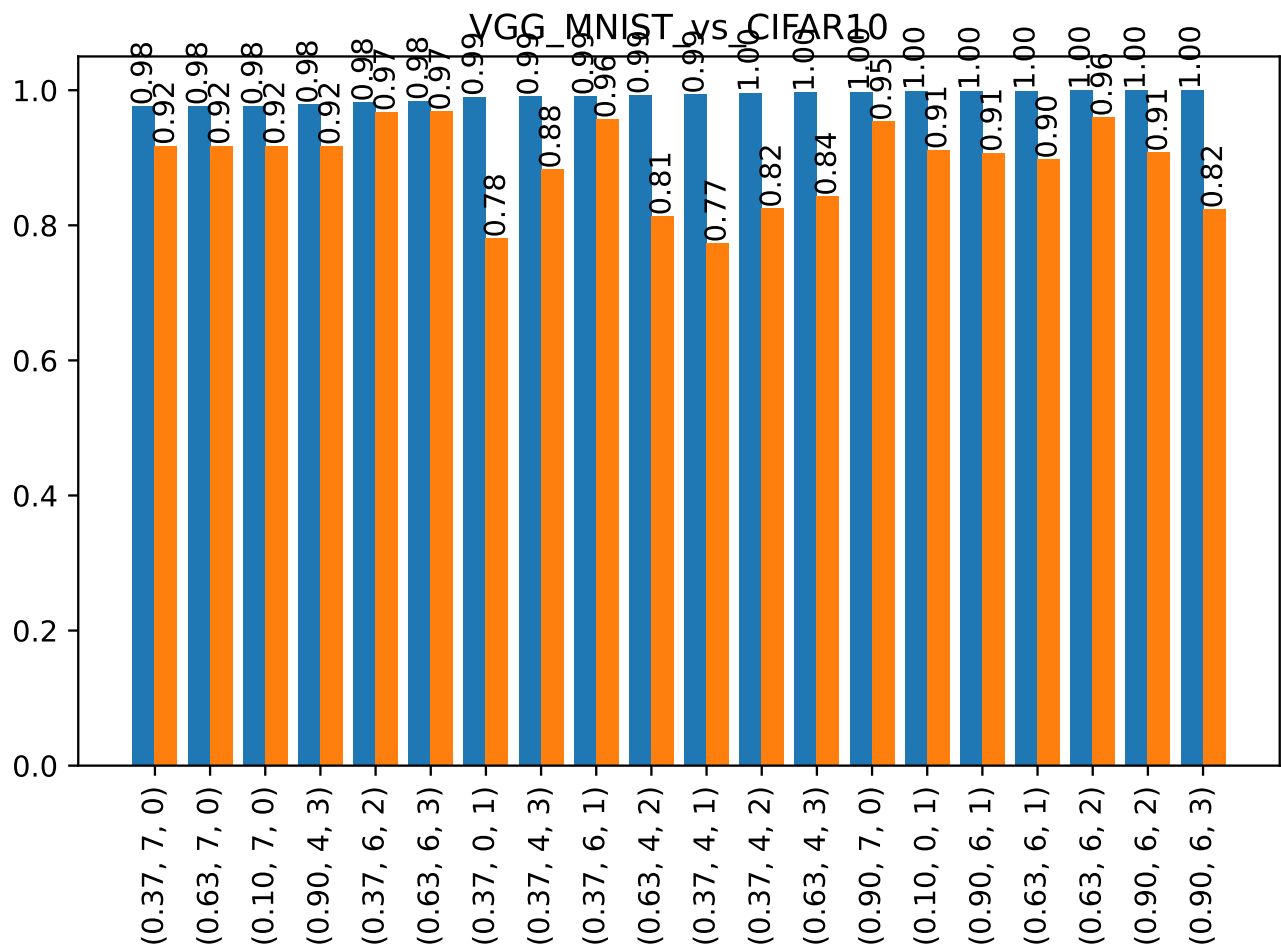


Figure 11: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

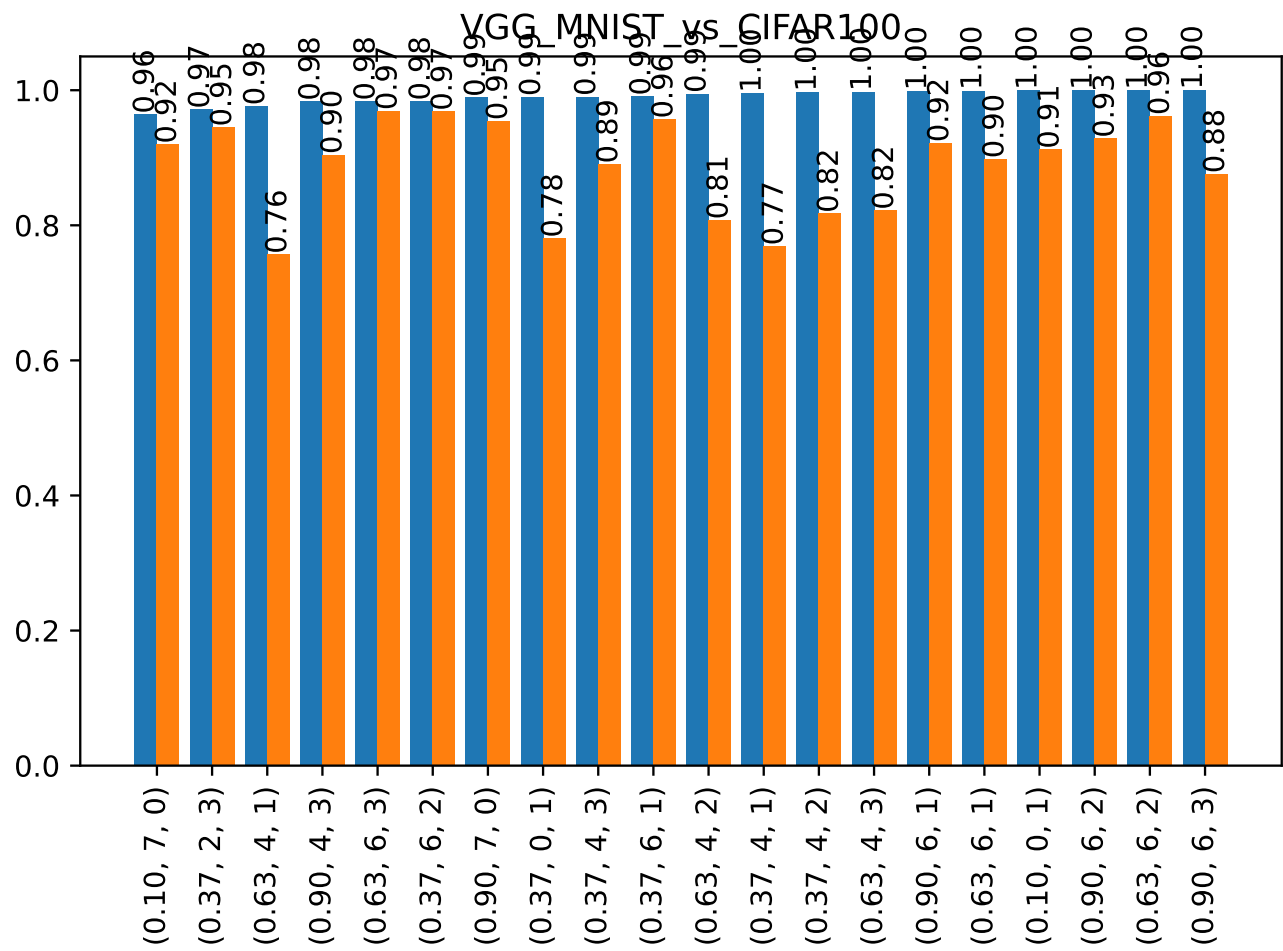


Figure 12: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

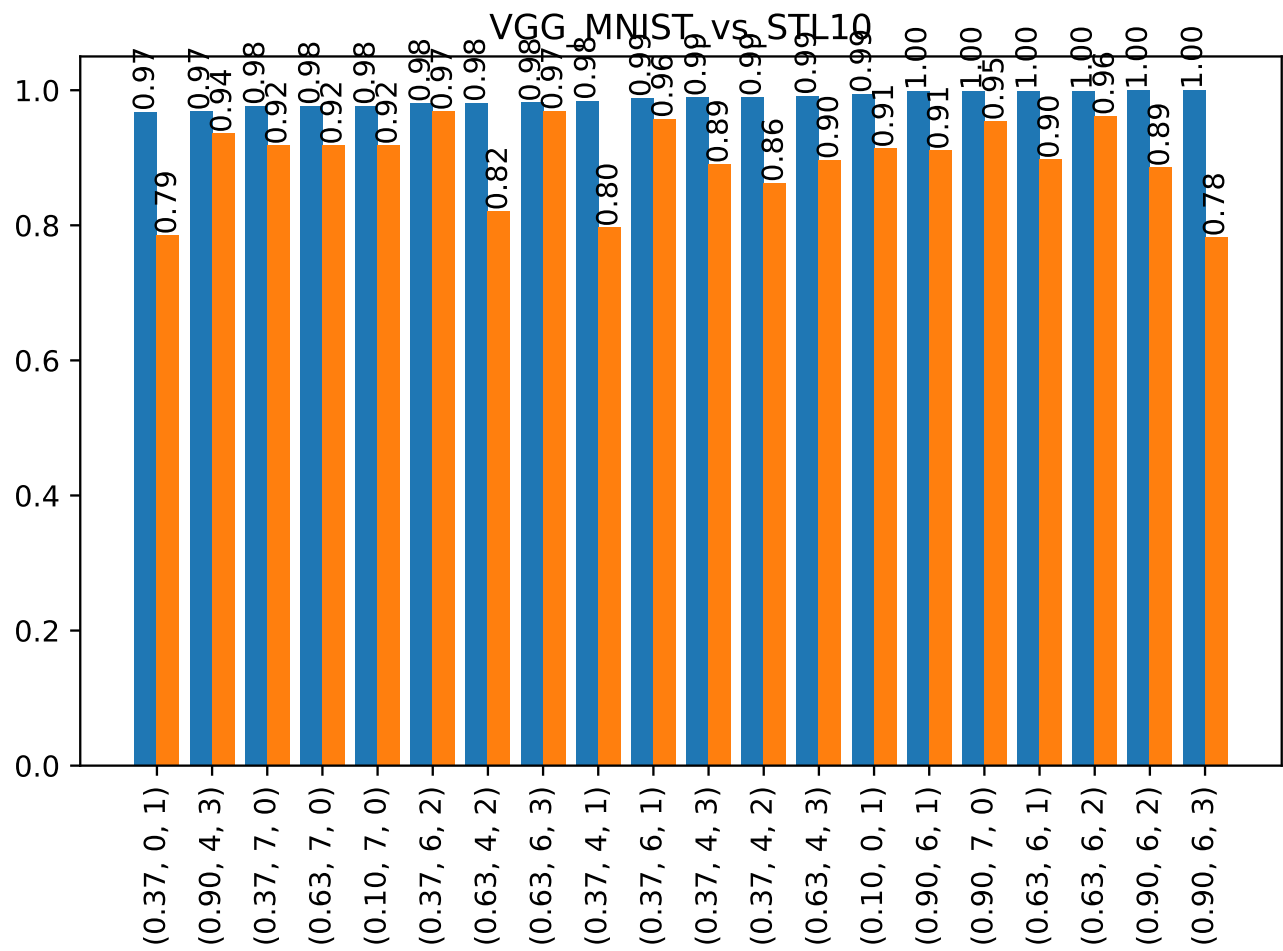


Figure 13: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

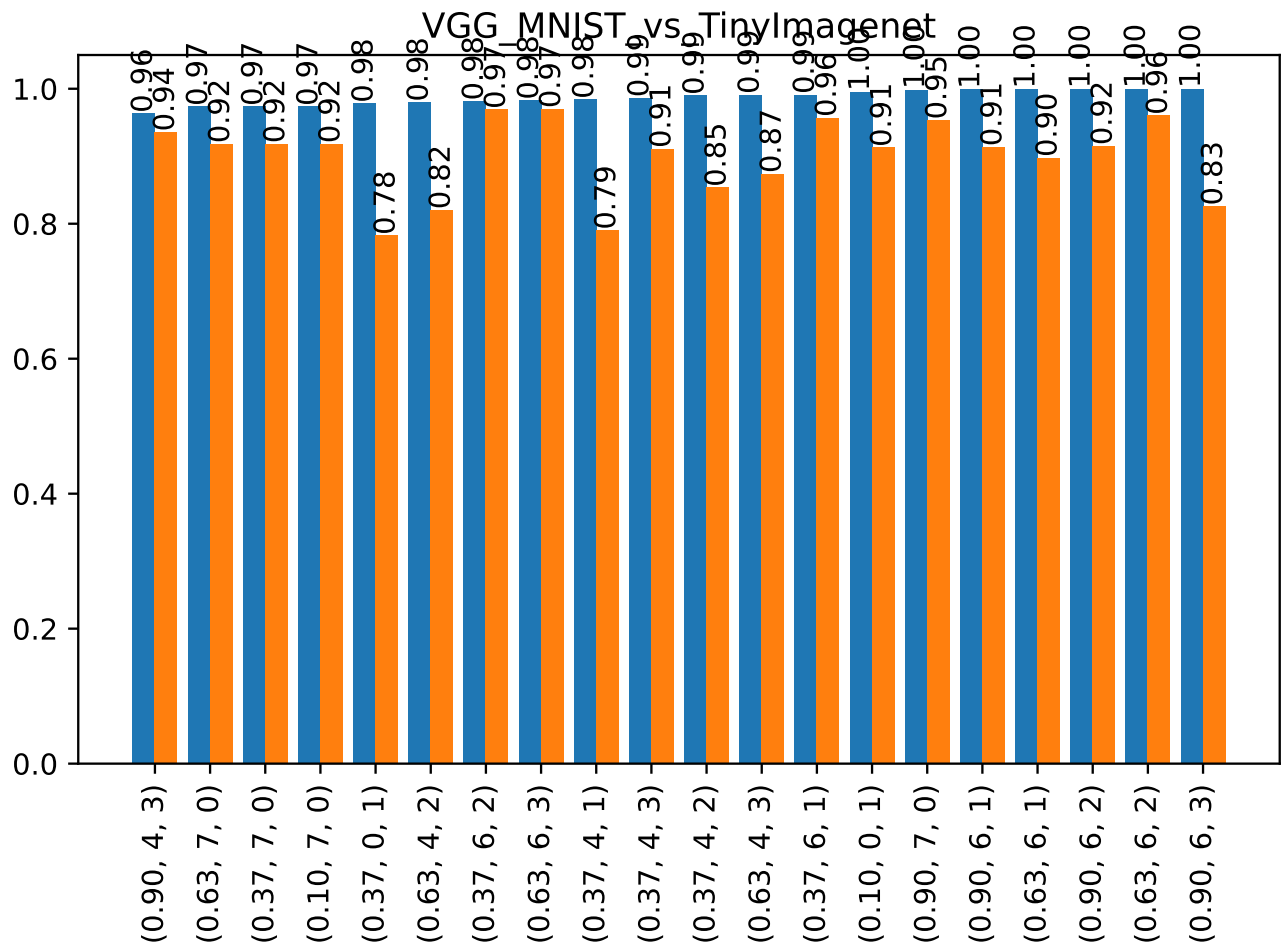


Figure 14: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

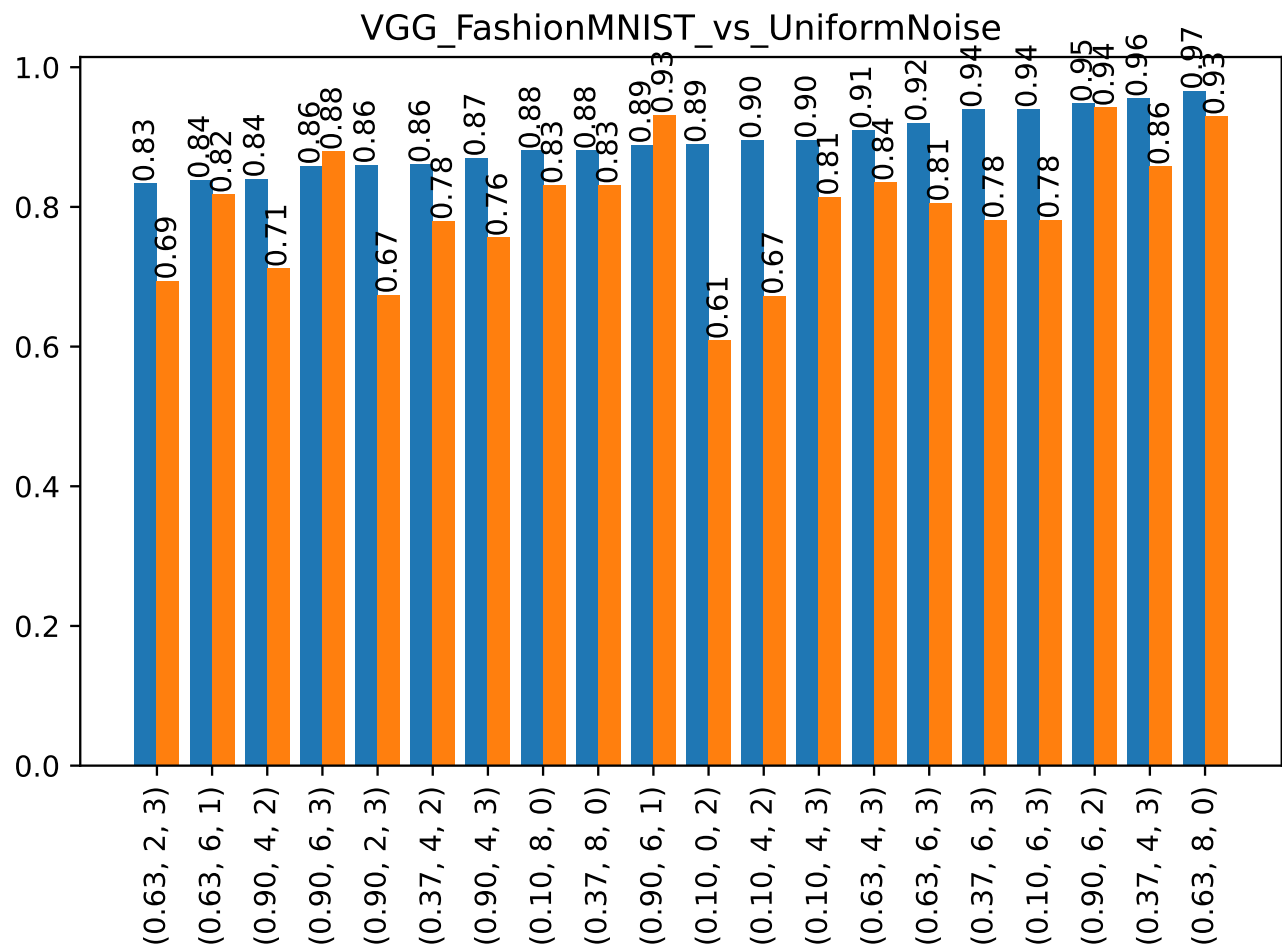


Figure 15: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

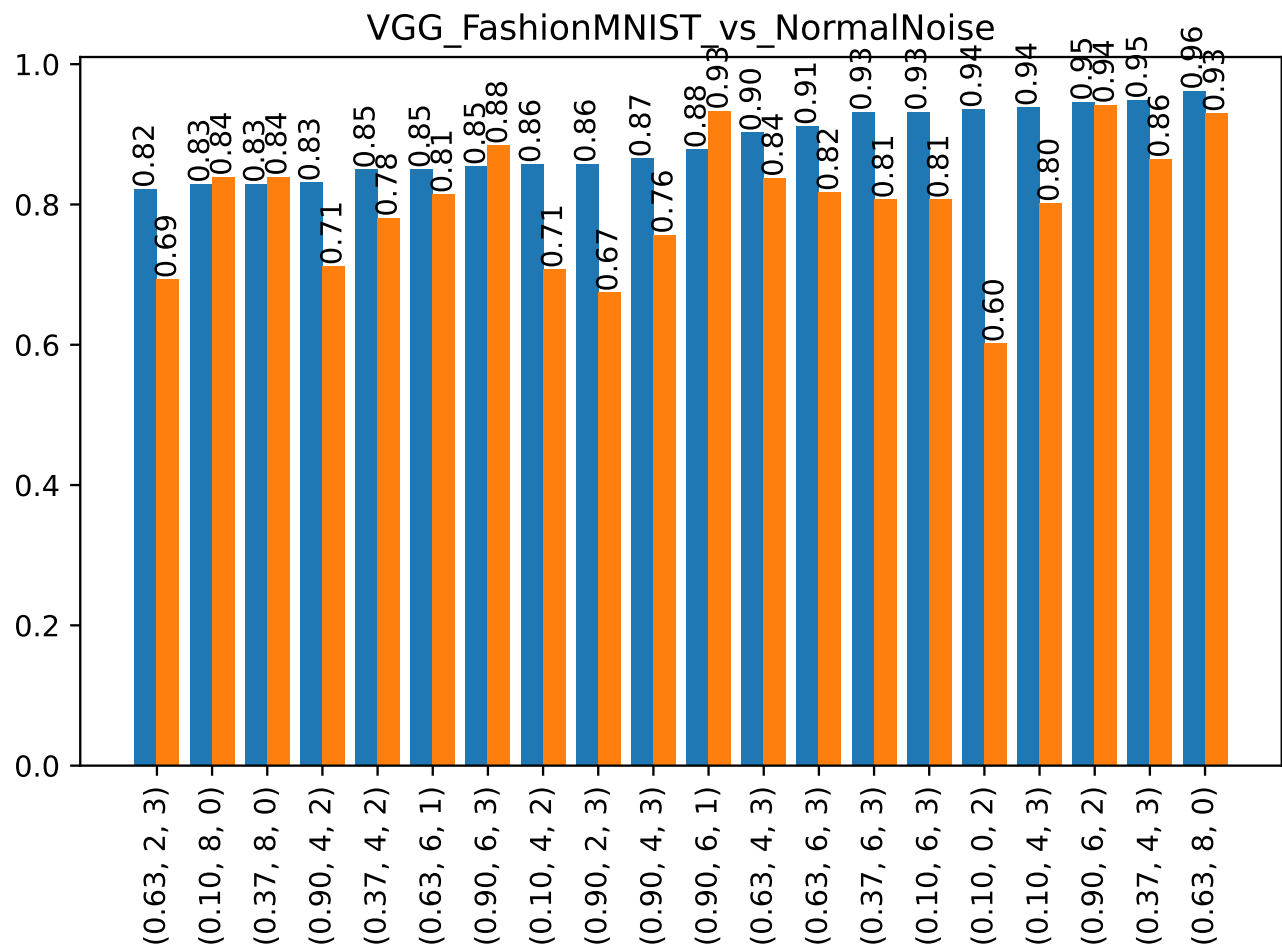


Figure 16: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaraneczowy).

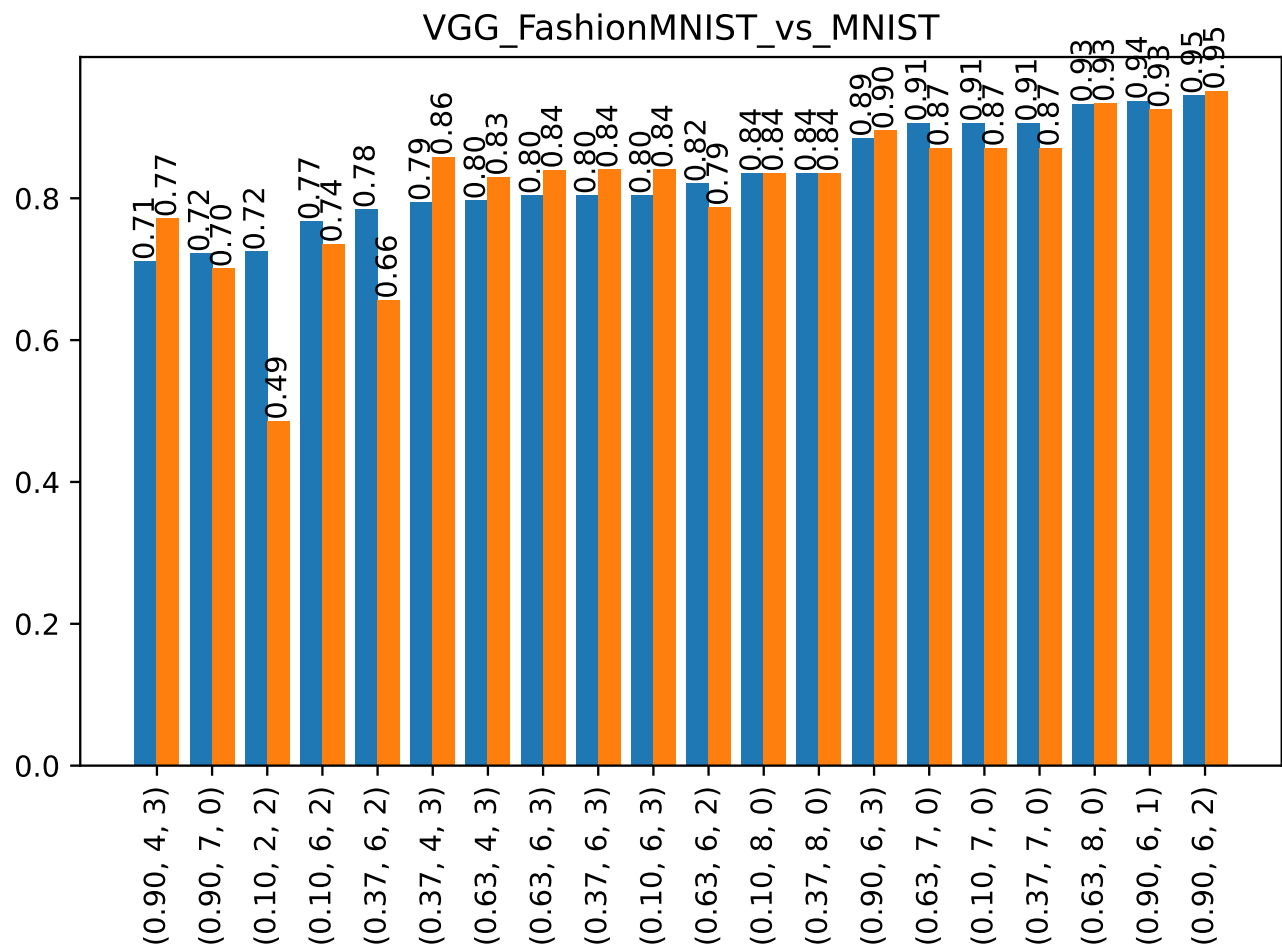


Figure 17: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

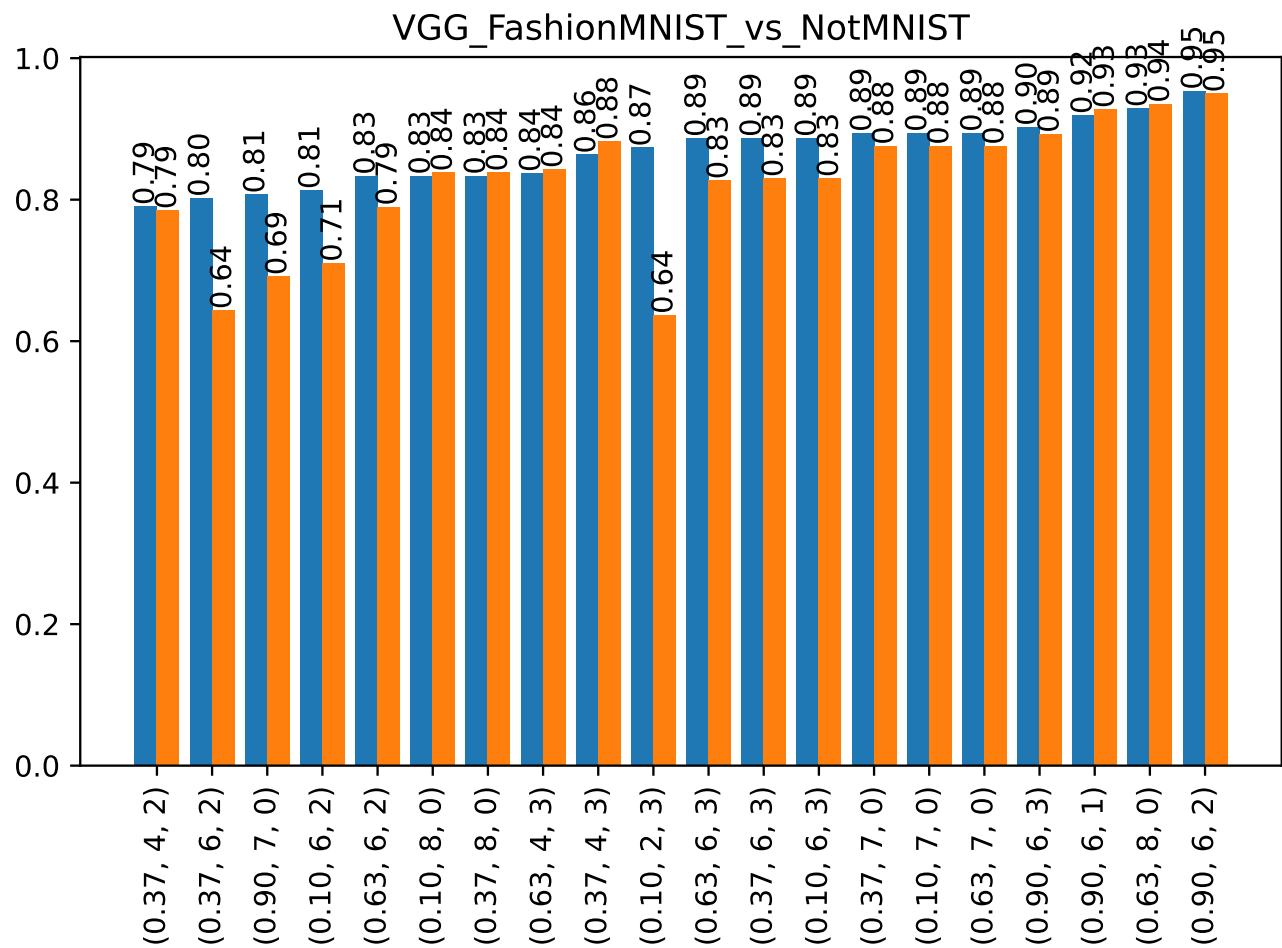


Figure 18: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

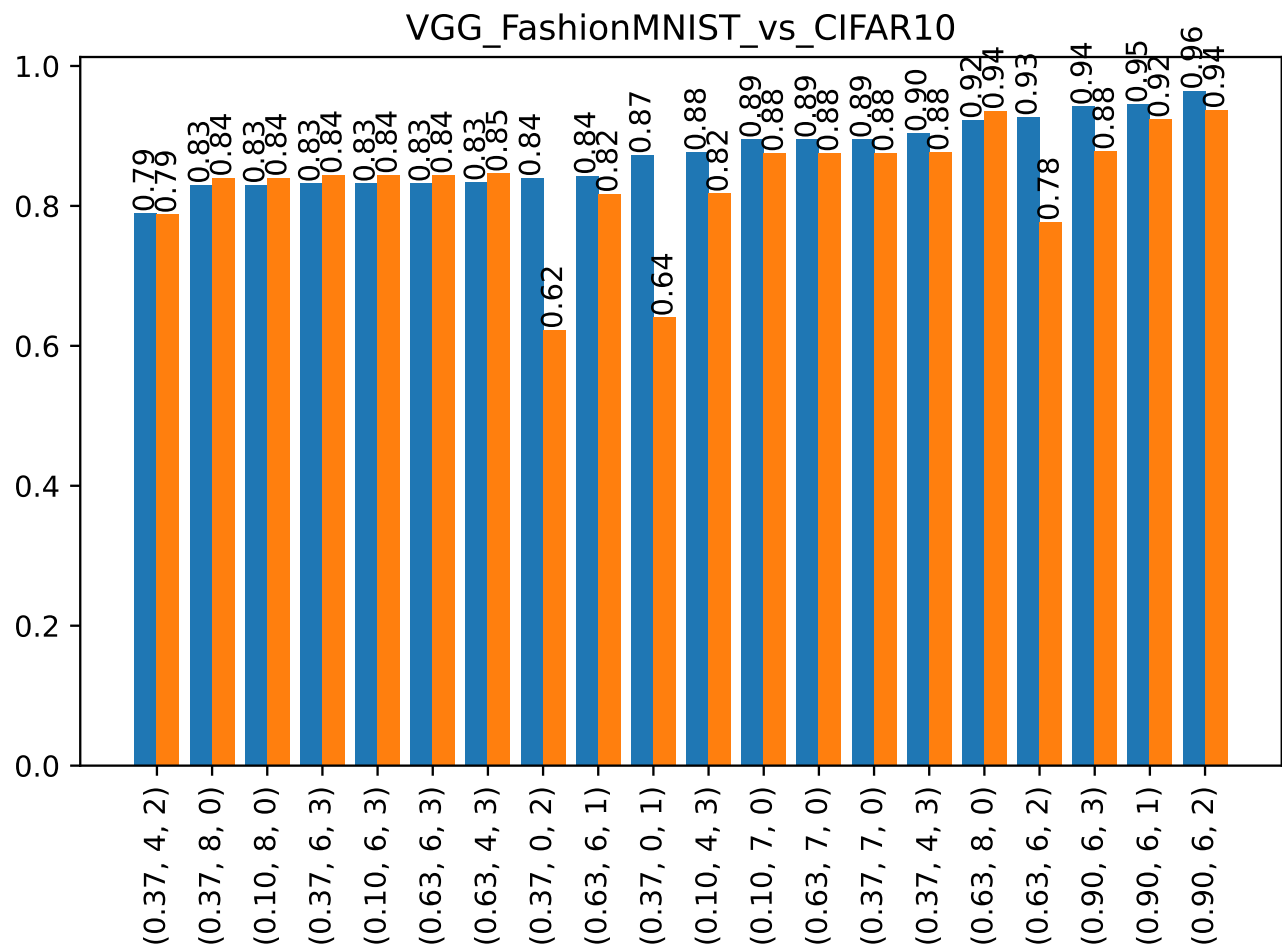


Figure 19: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

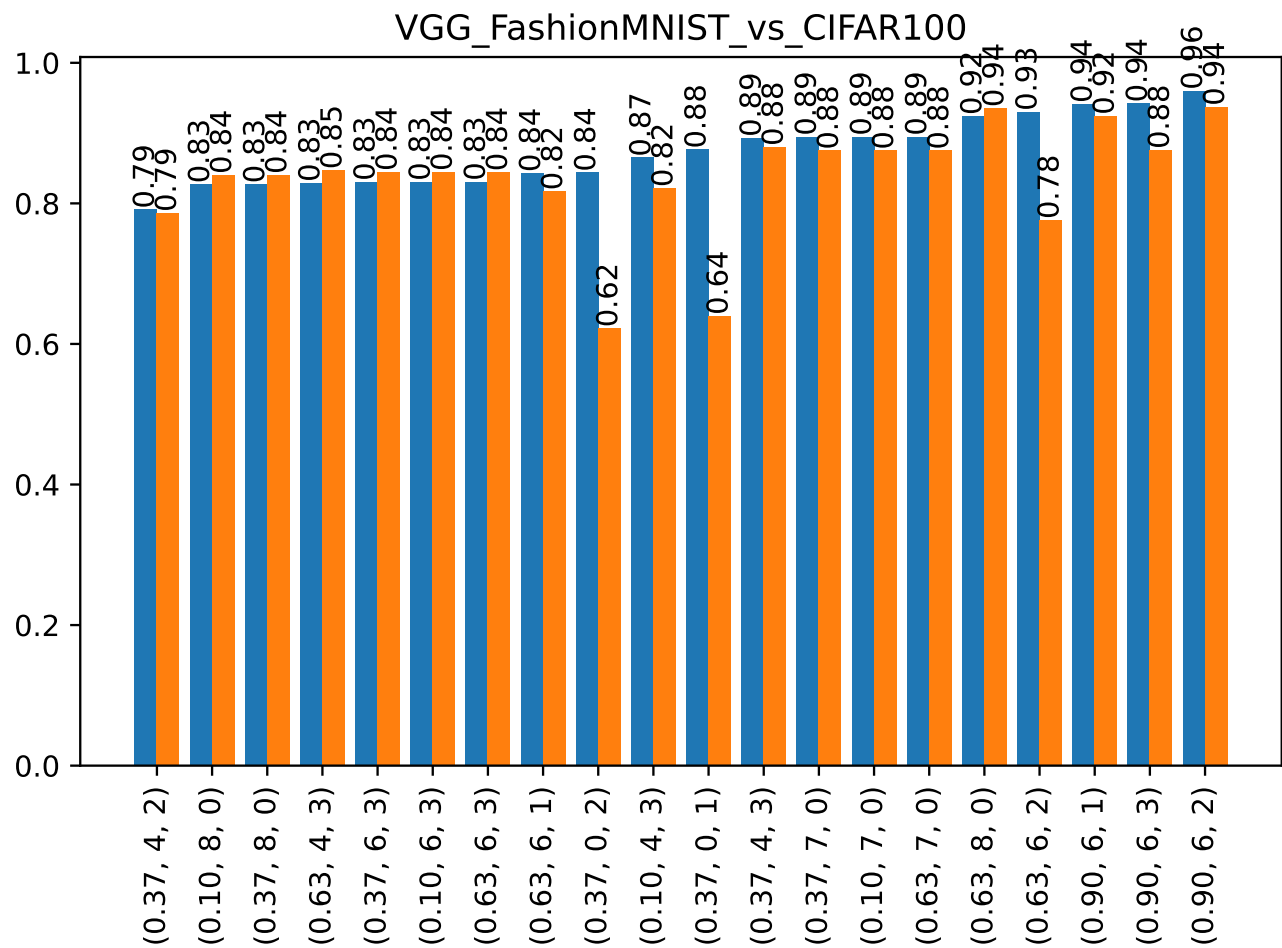


Figure 20: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

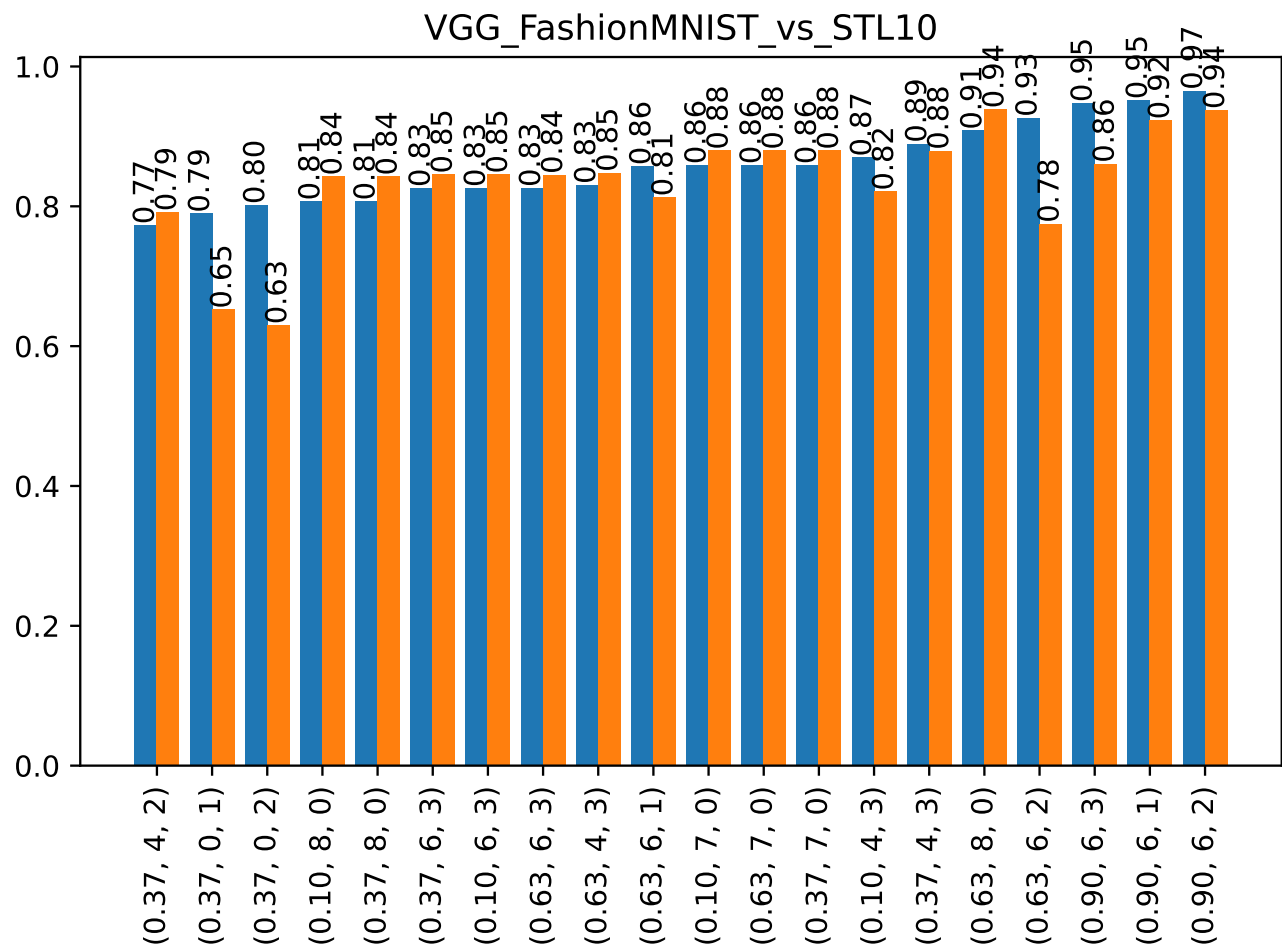


Figure 21: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

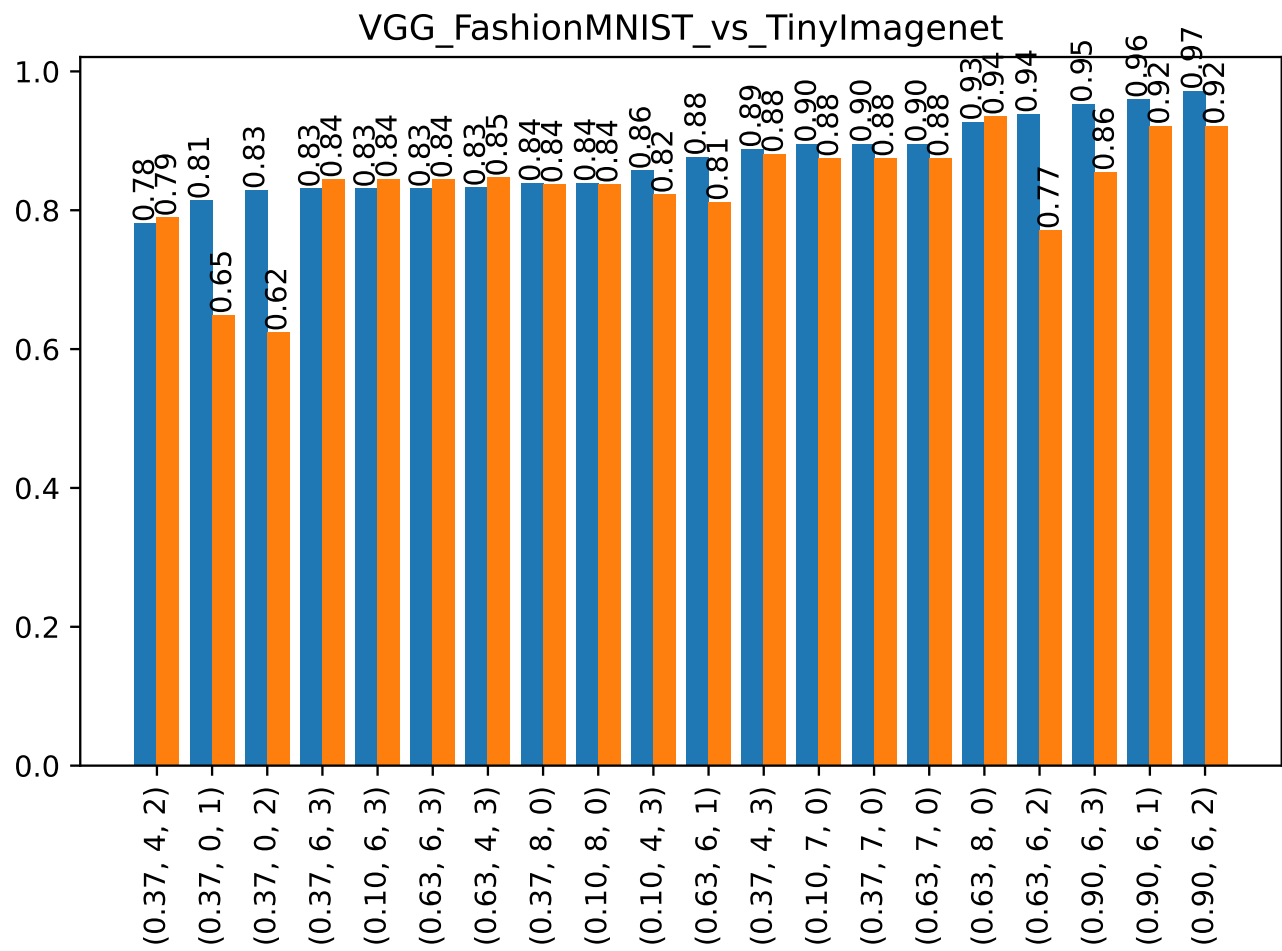


Figure 22: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

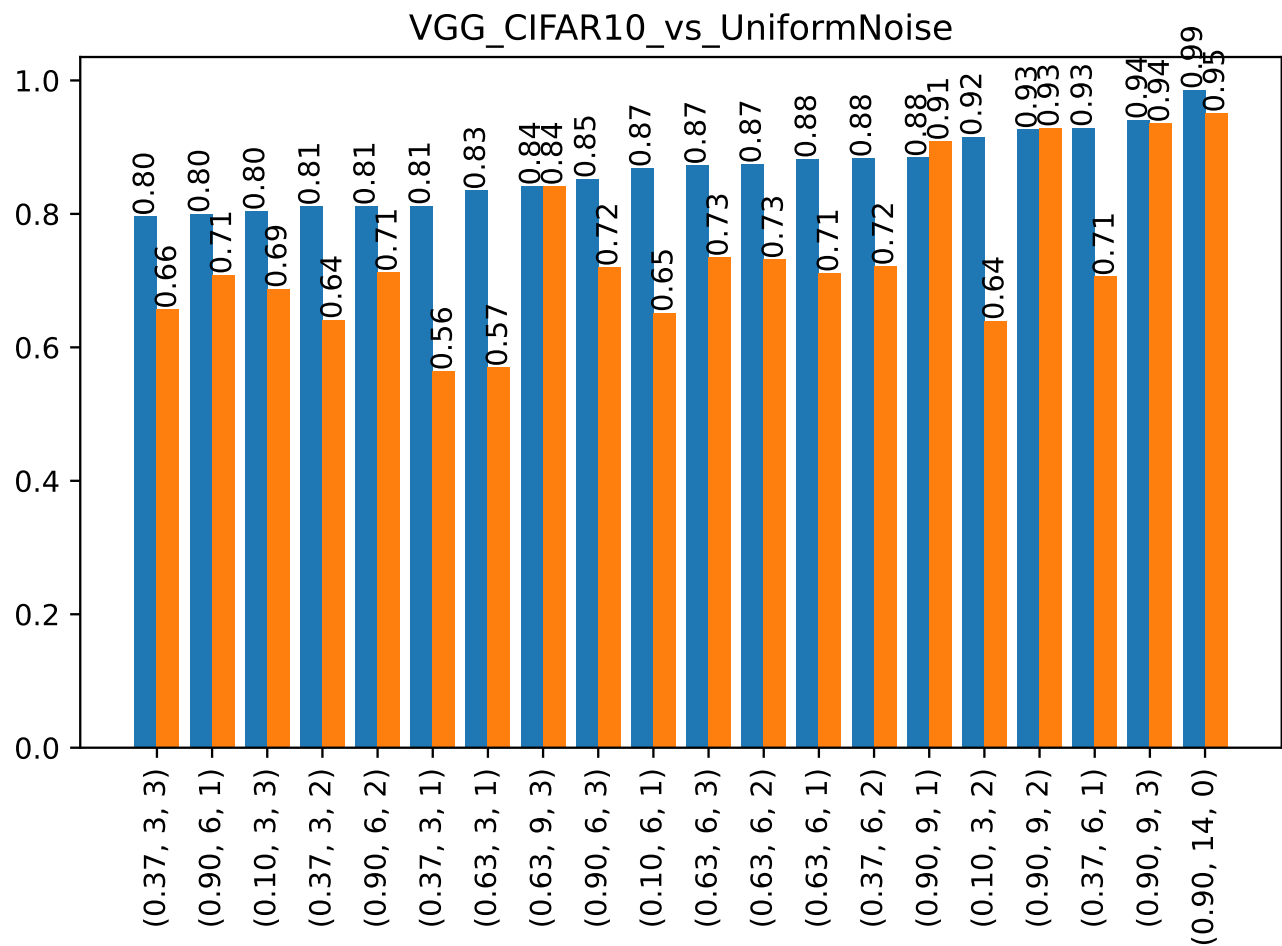


Figure 23: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

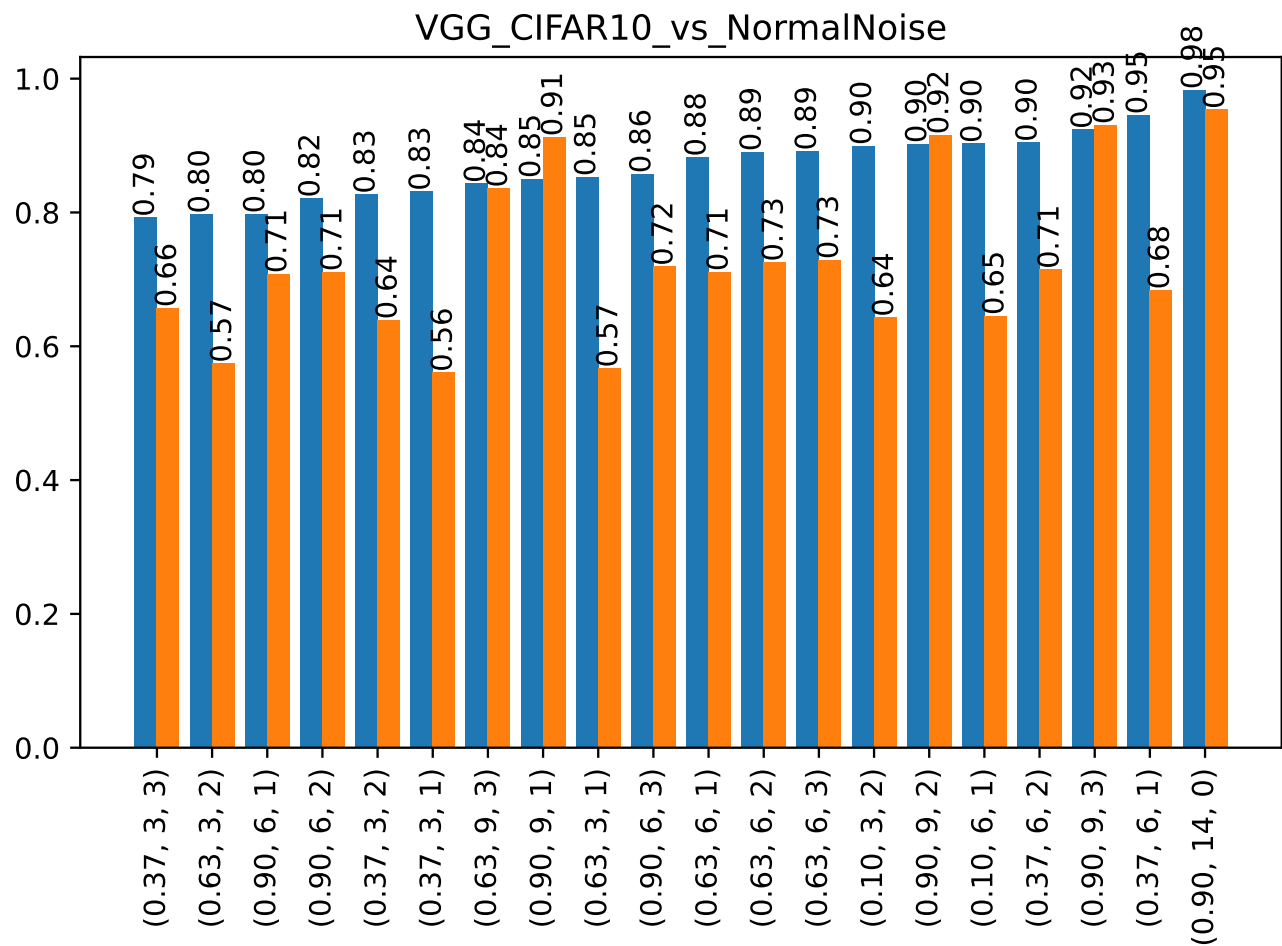


Figure 24: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

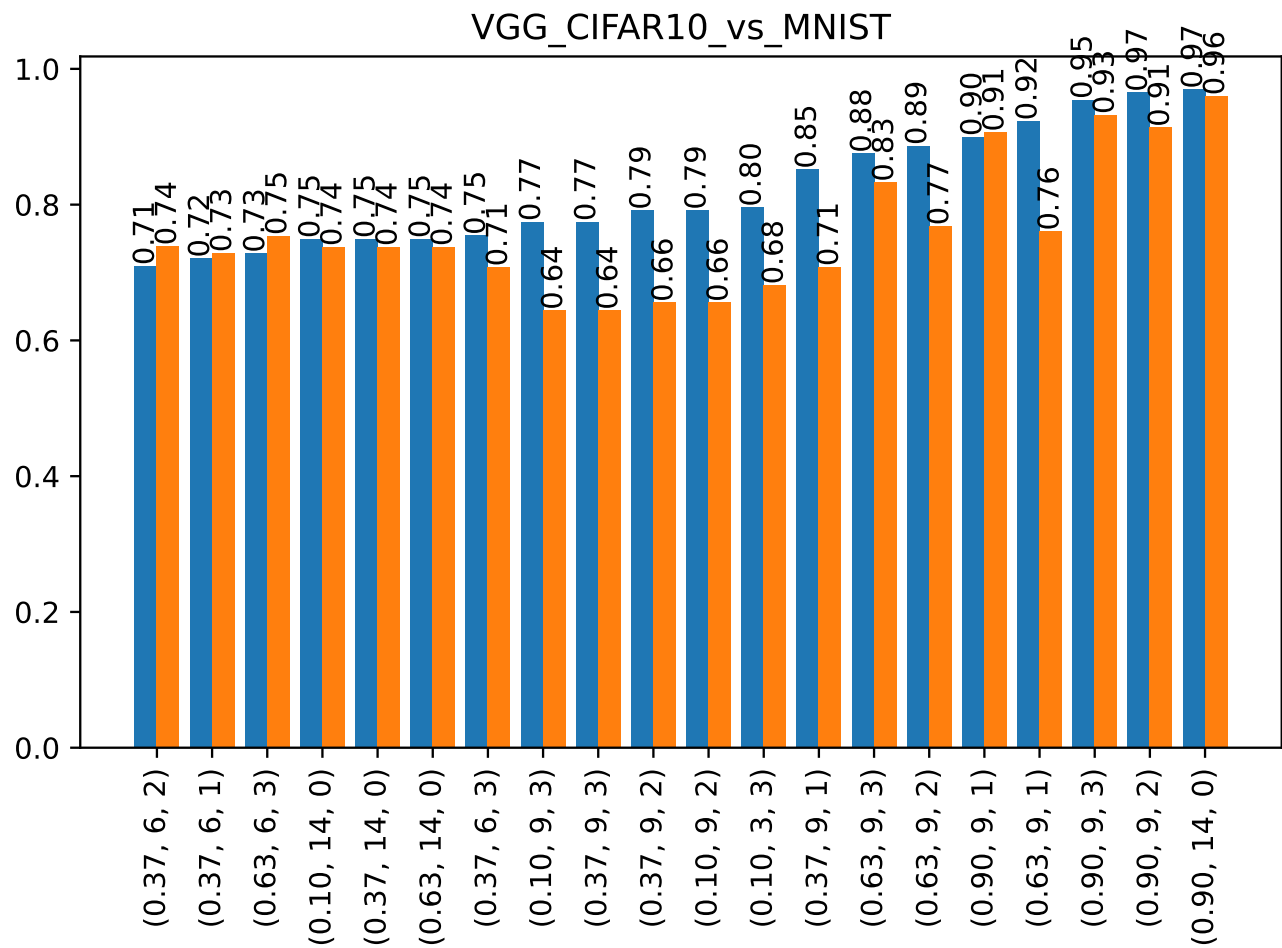


Figure 25: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

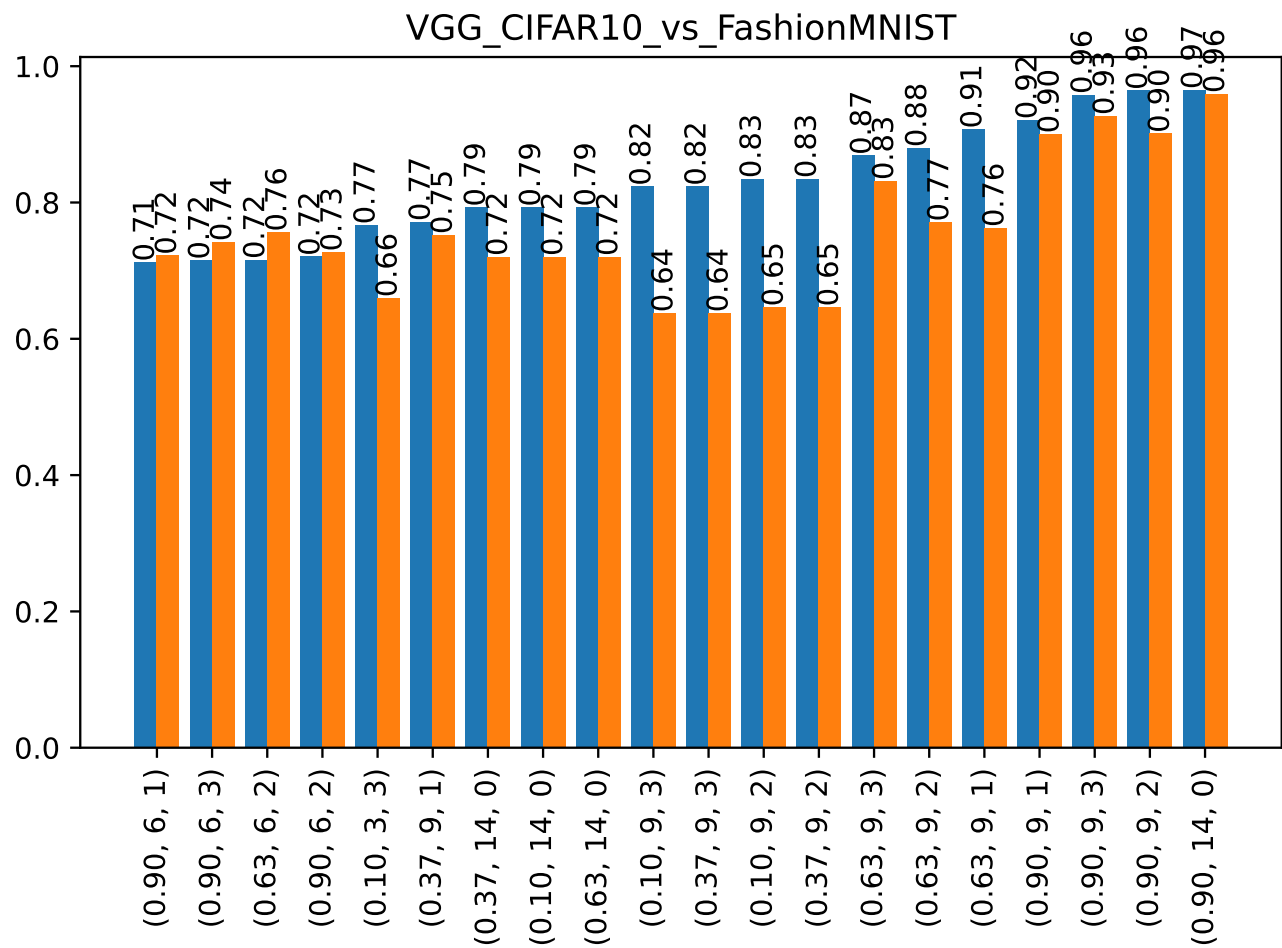


Figure 26: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

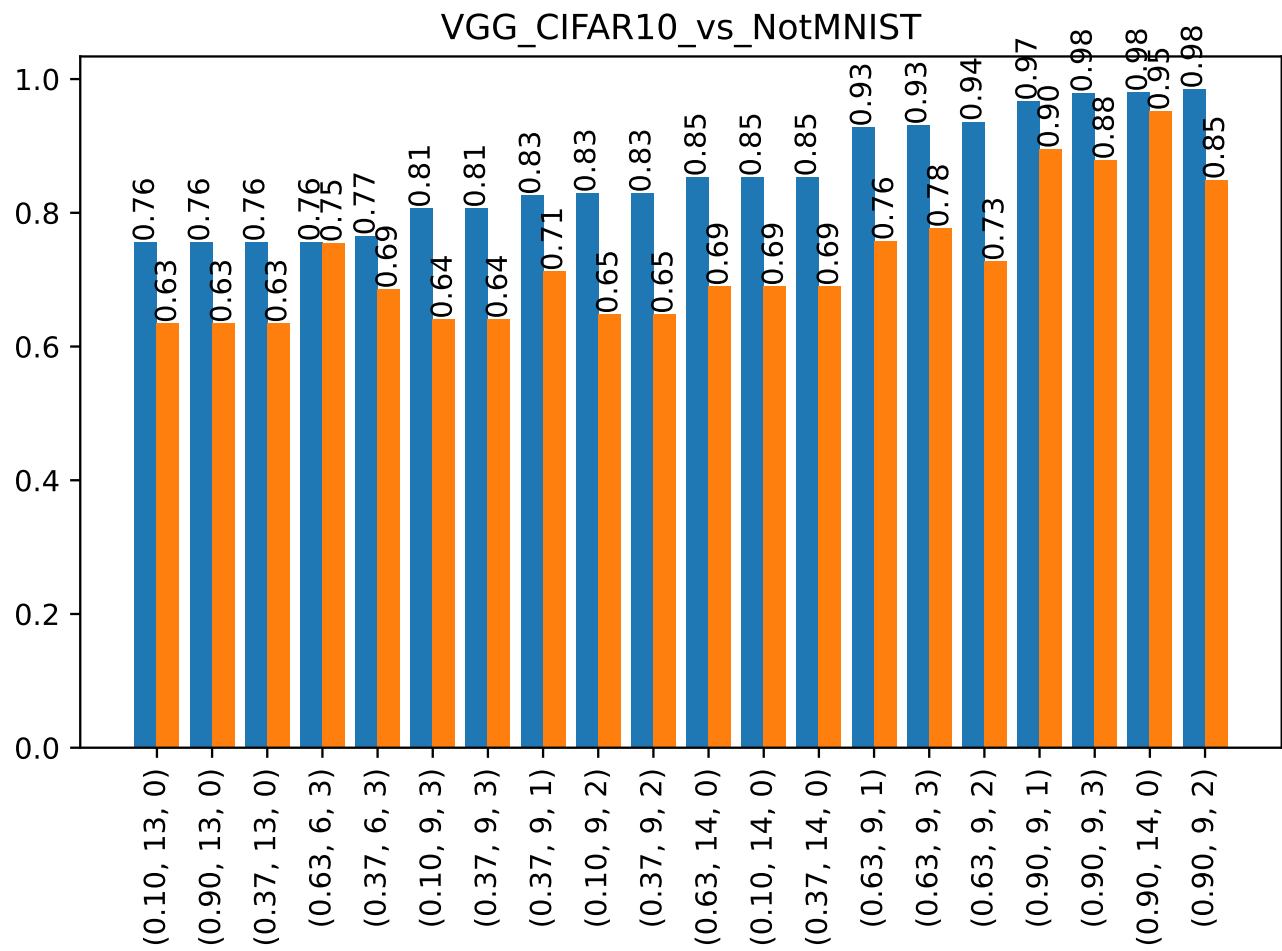


Figure 27: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

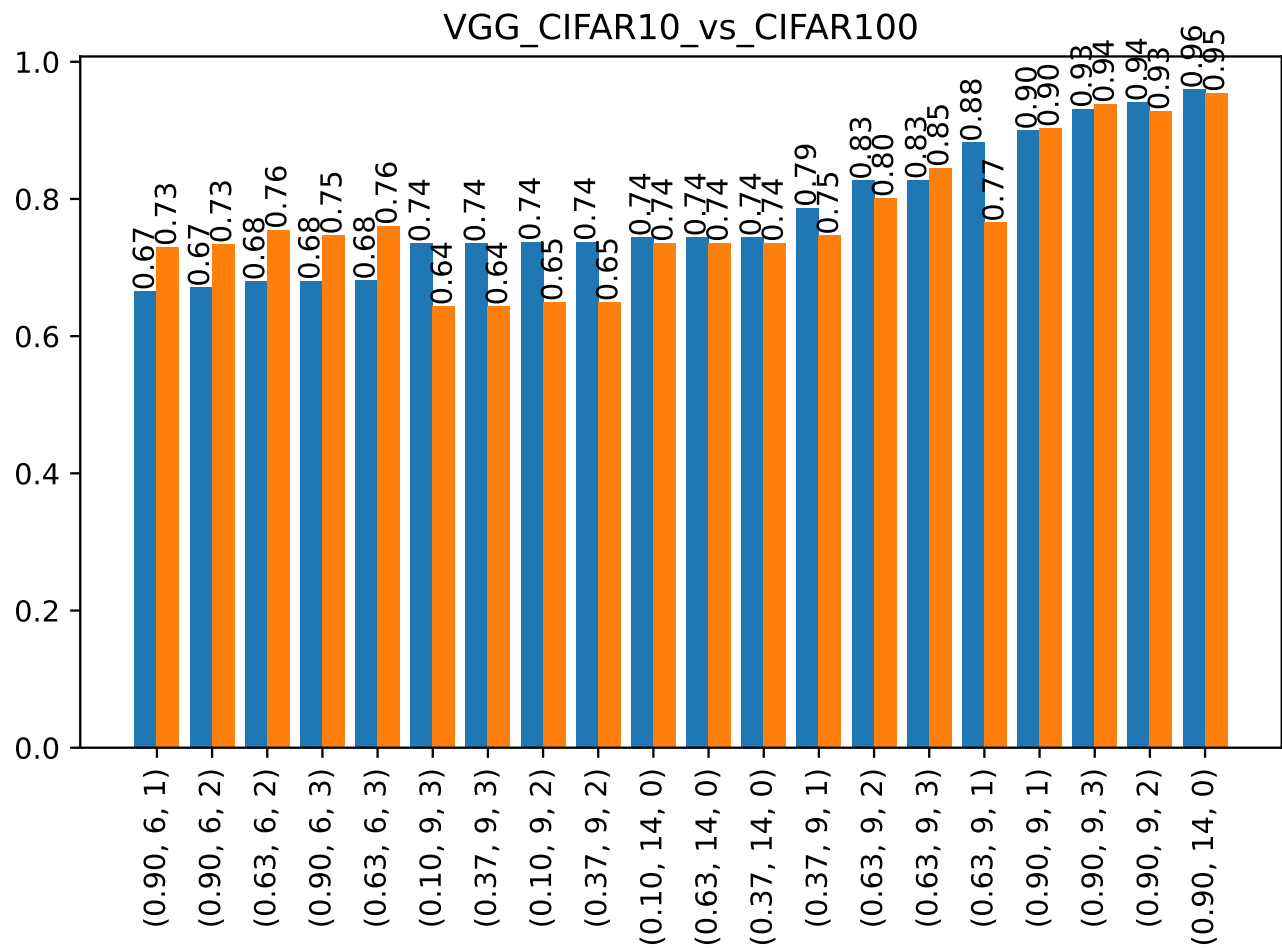


Figure 28: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

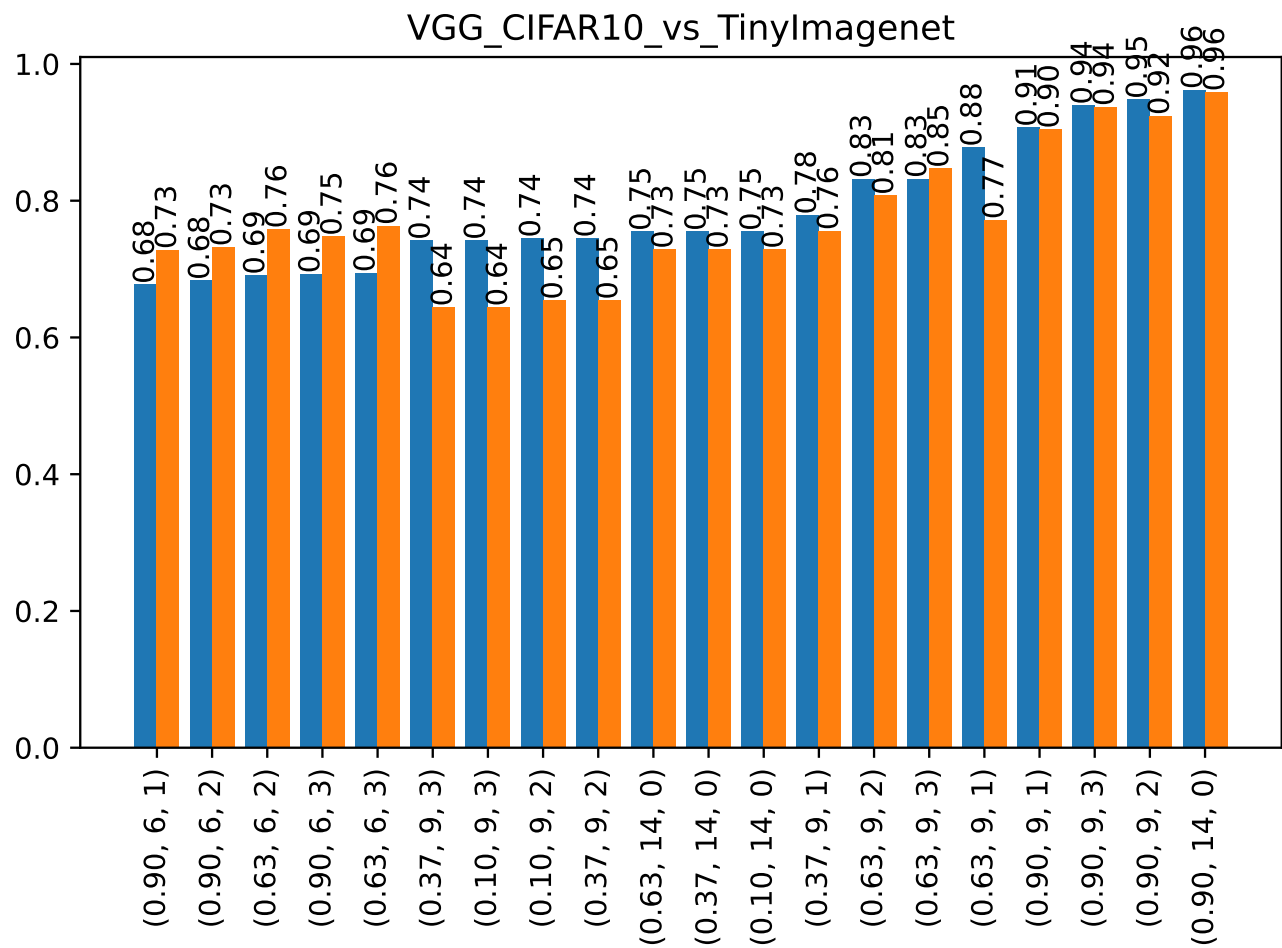


Figure 29: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

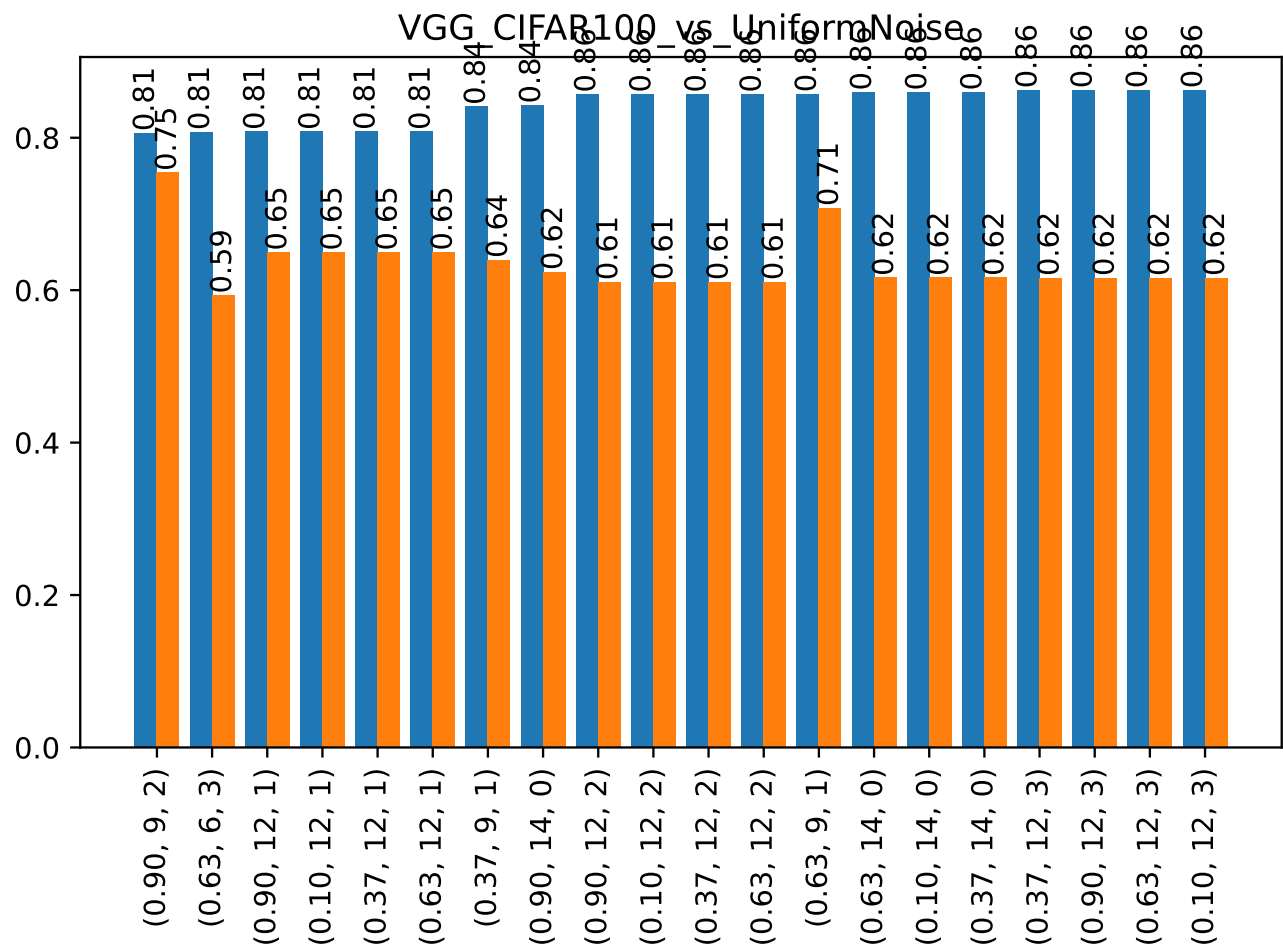


Figure 30: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaraneczowy).

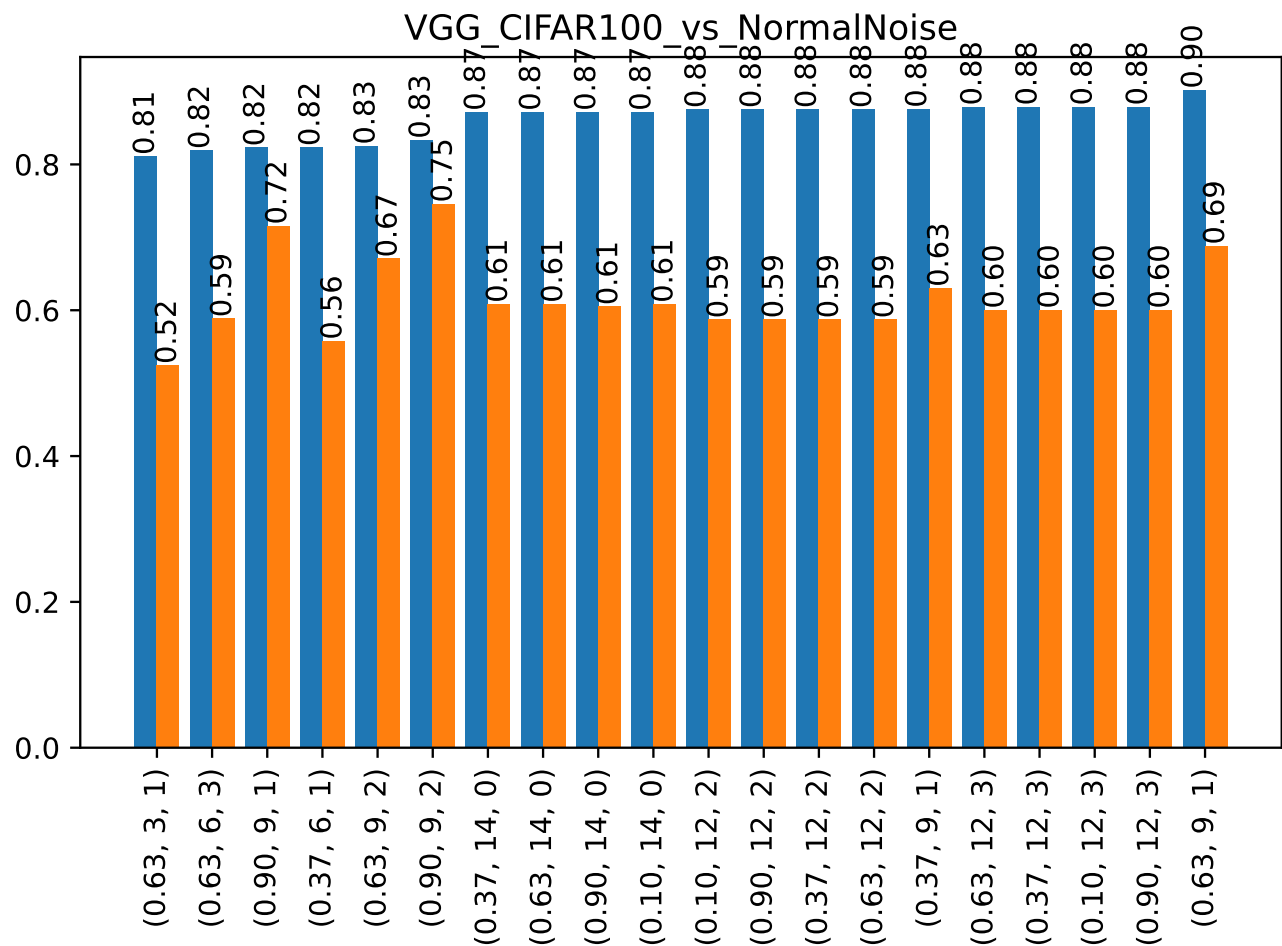


Figure 31: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

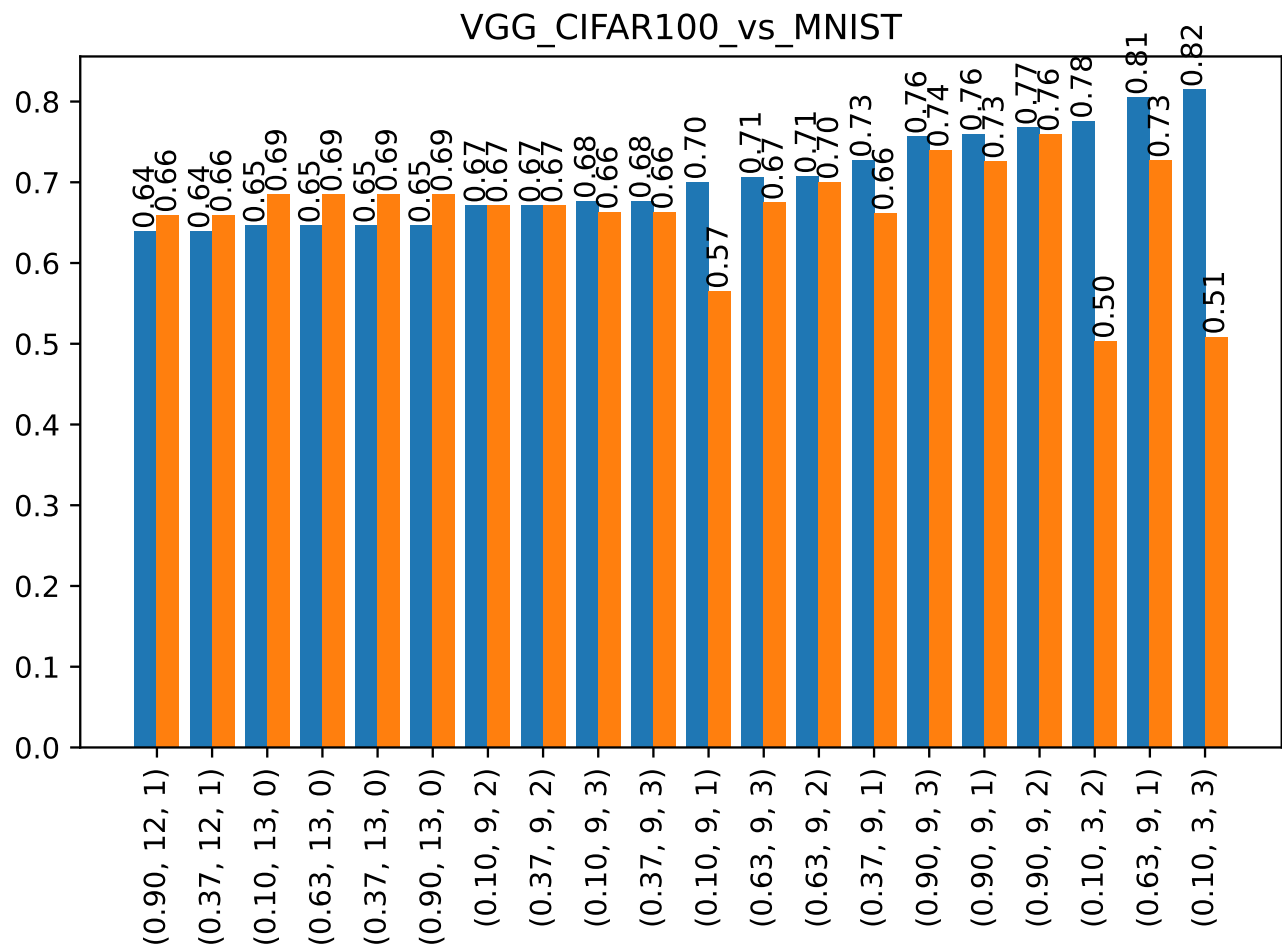


Figure 32: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

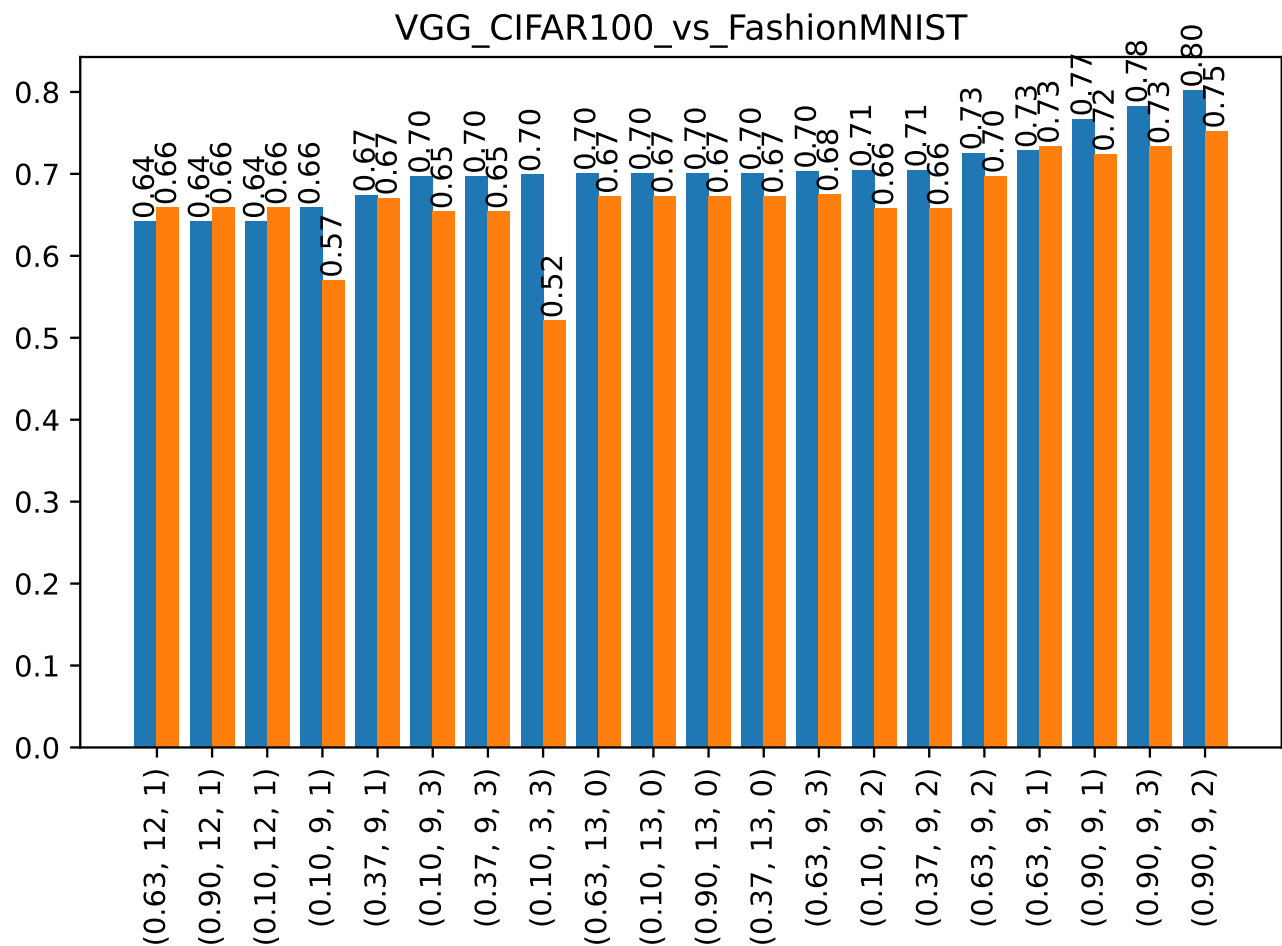


Figure 33: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

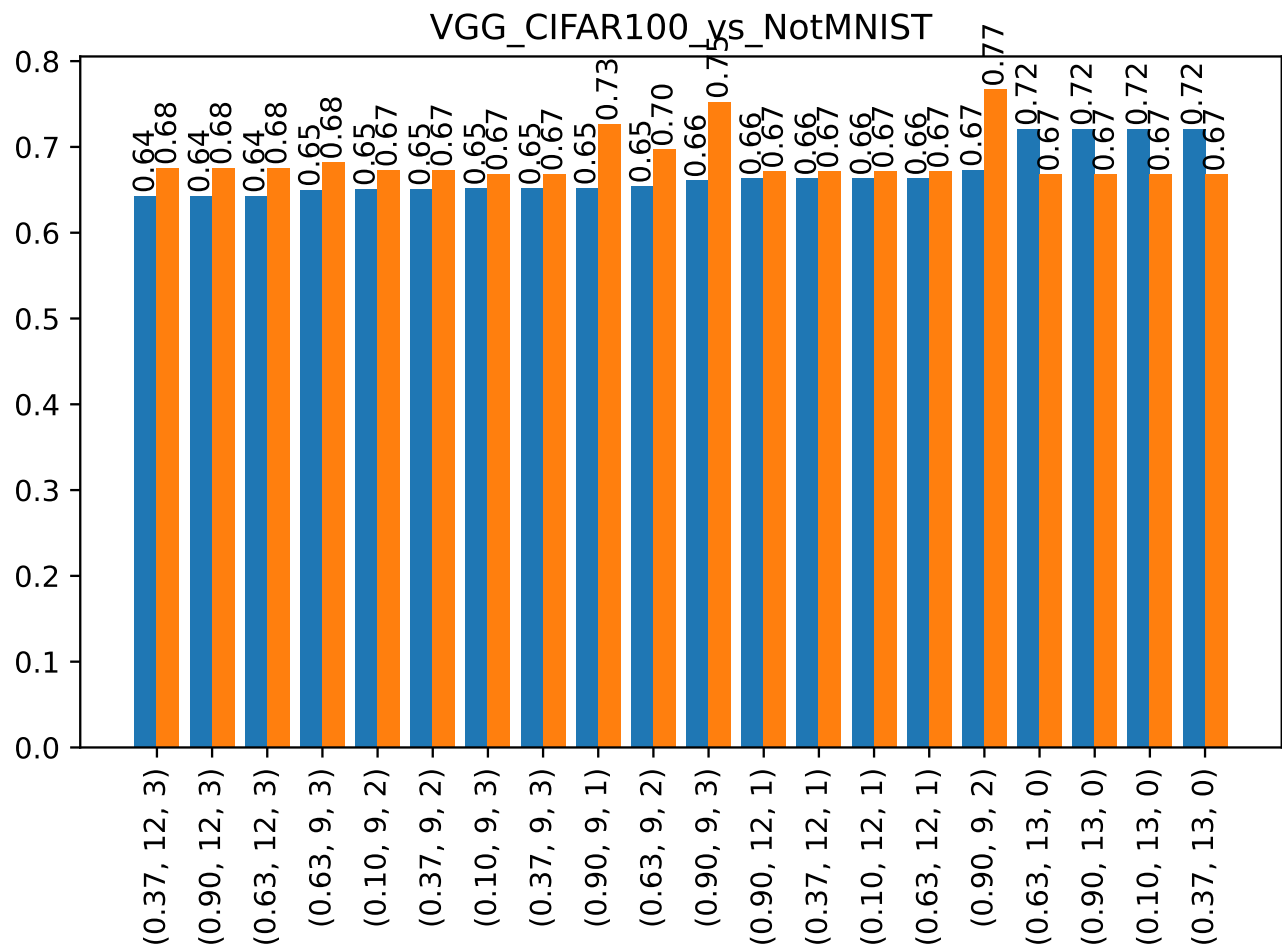


Figure 34: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

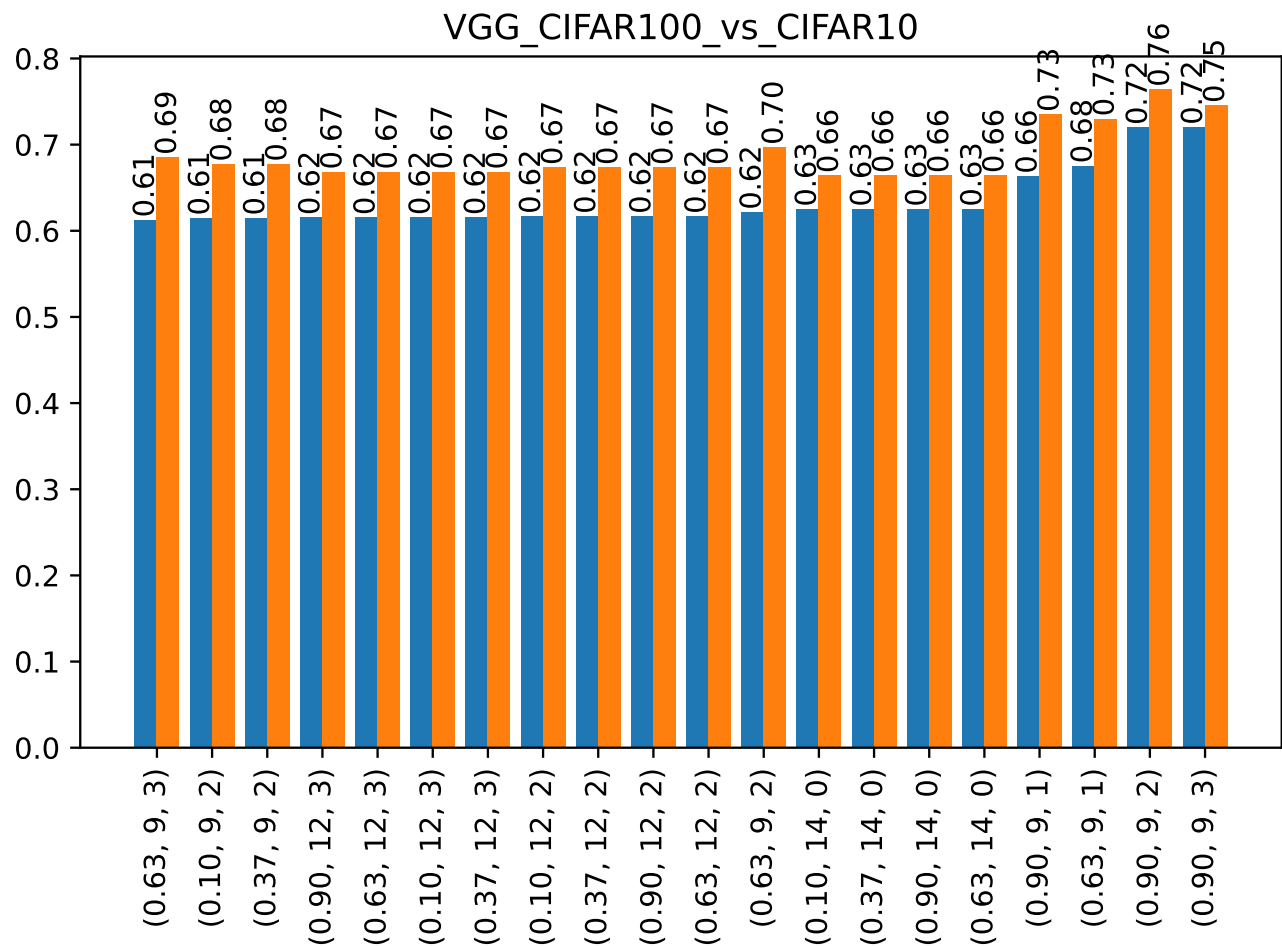


Figure 35: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

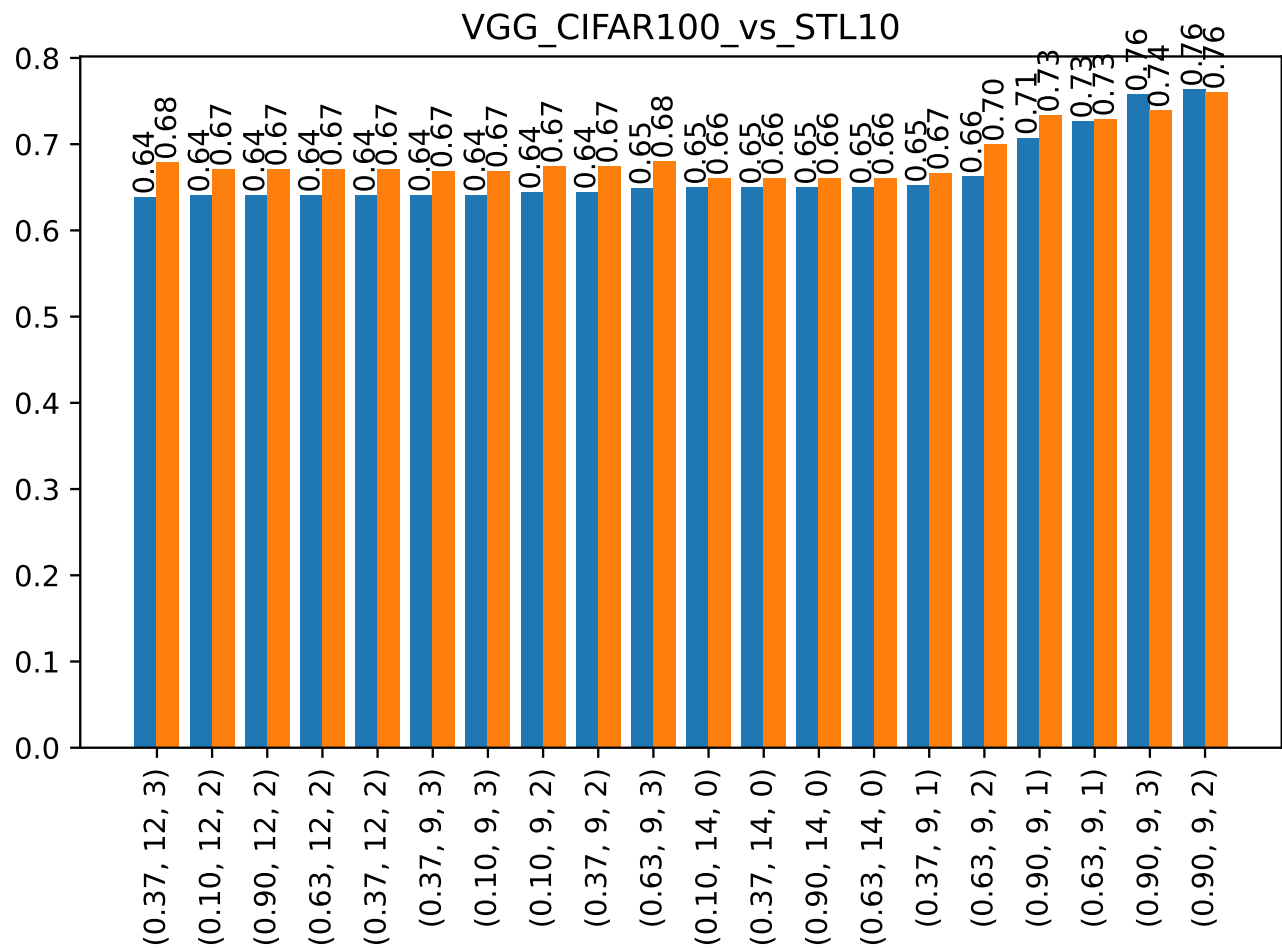


Figure 36: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

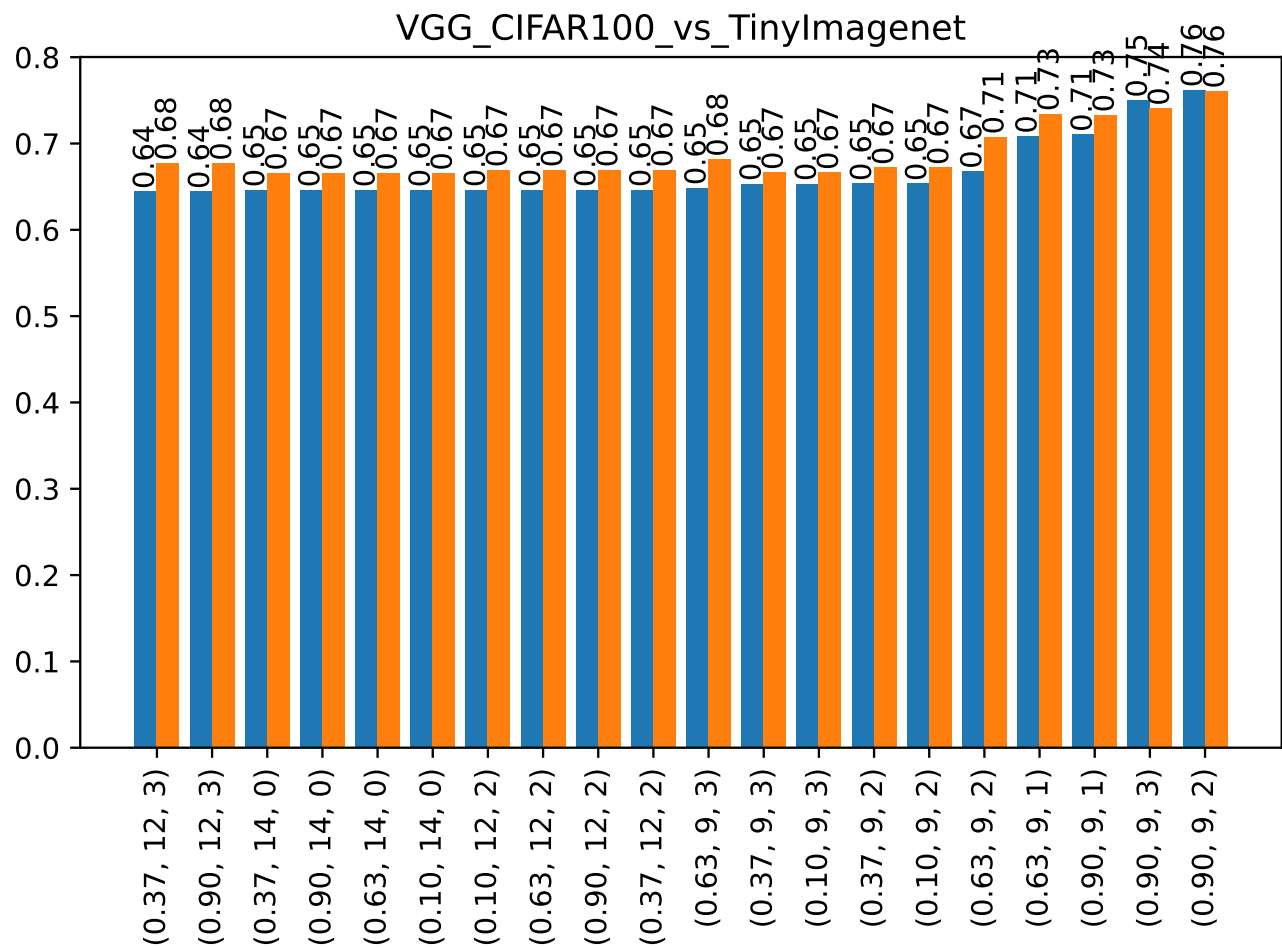


Figure 37: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

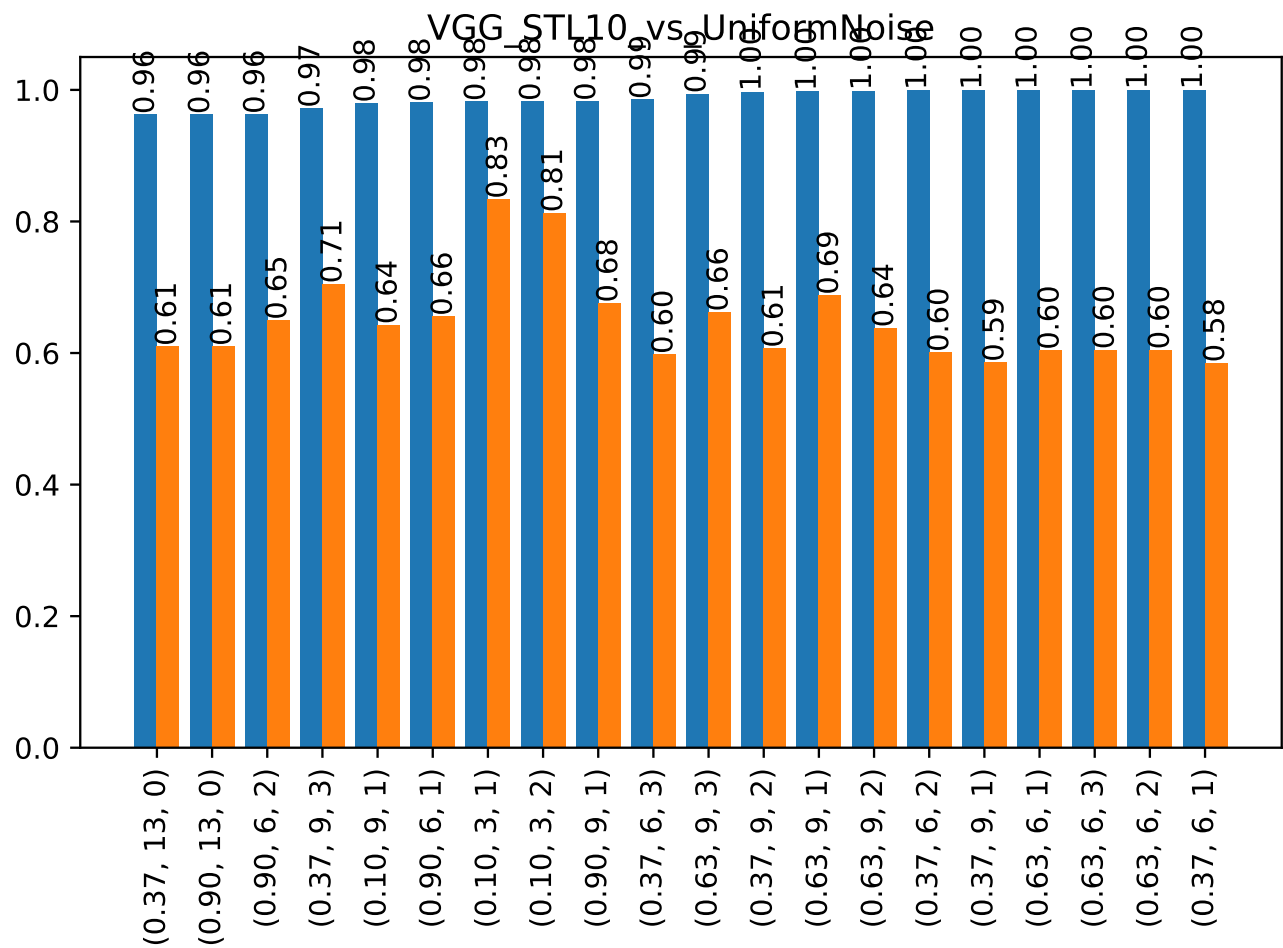


Figure 38: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

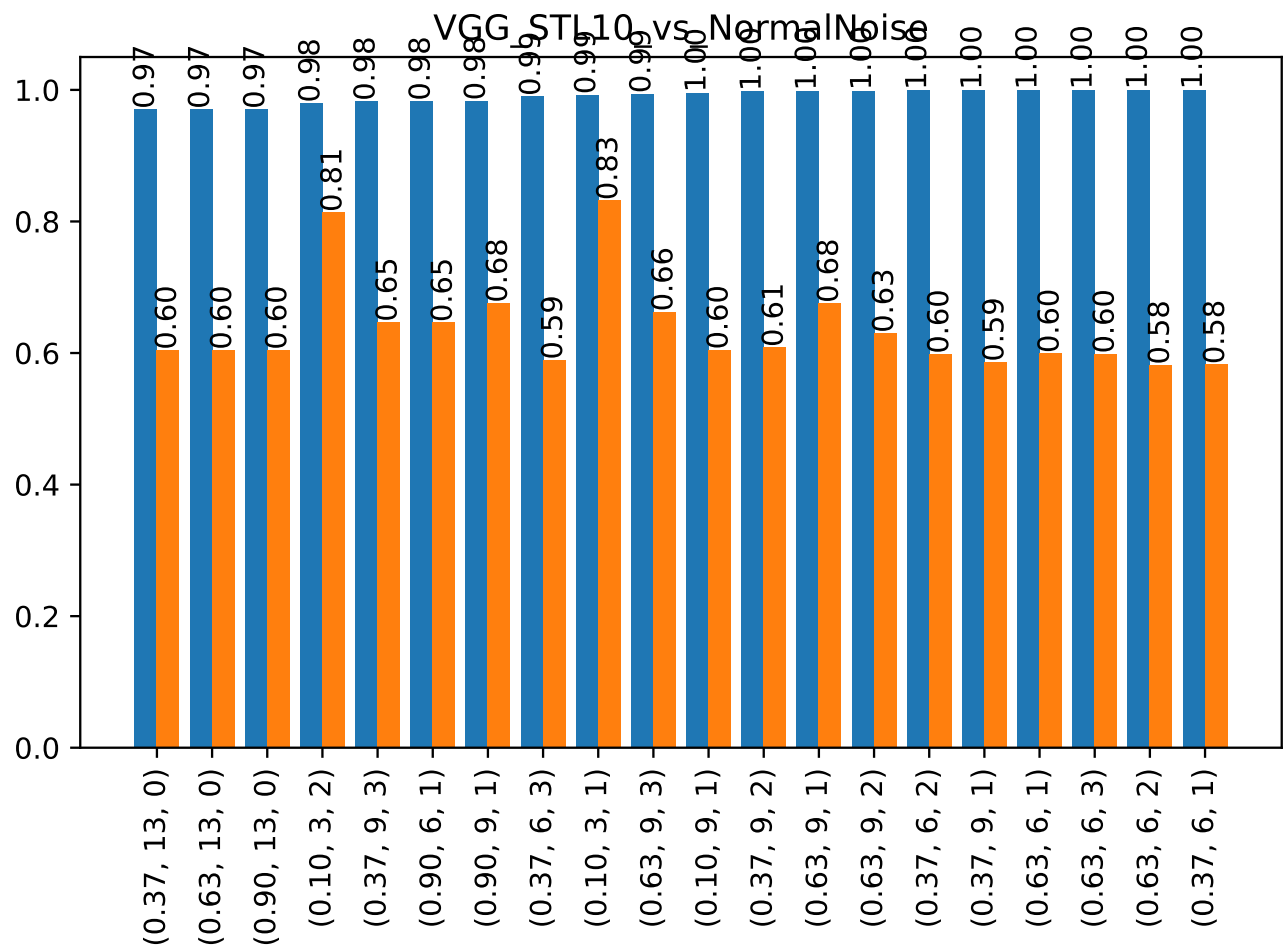


Figure 39: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

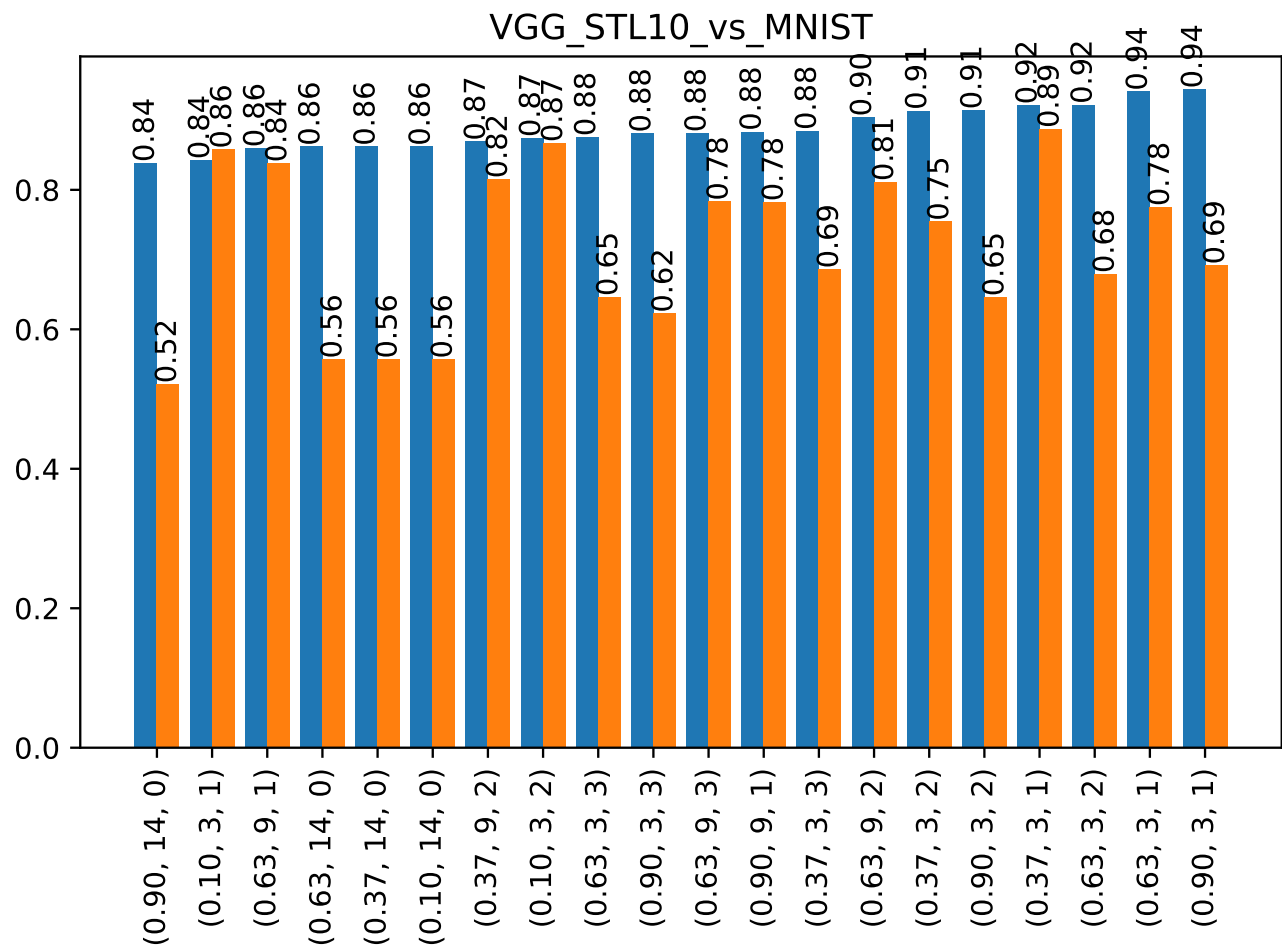


Figure 40: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

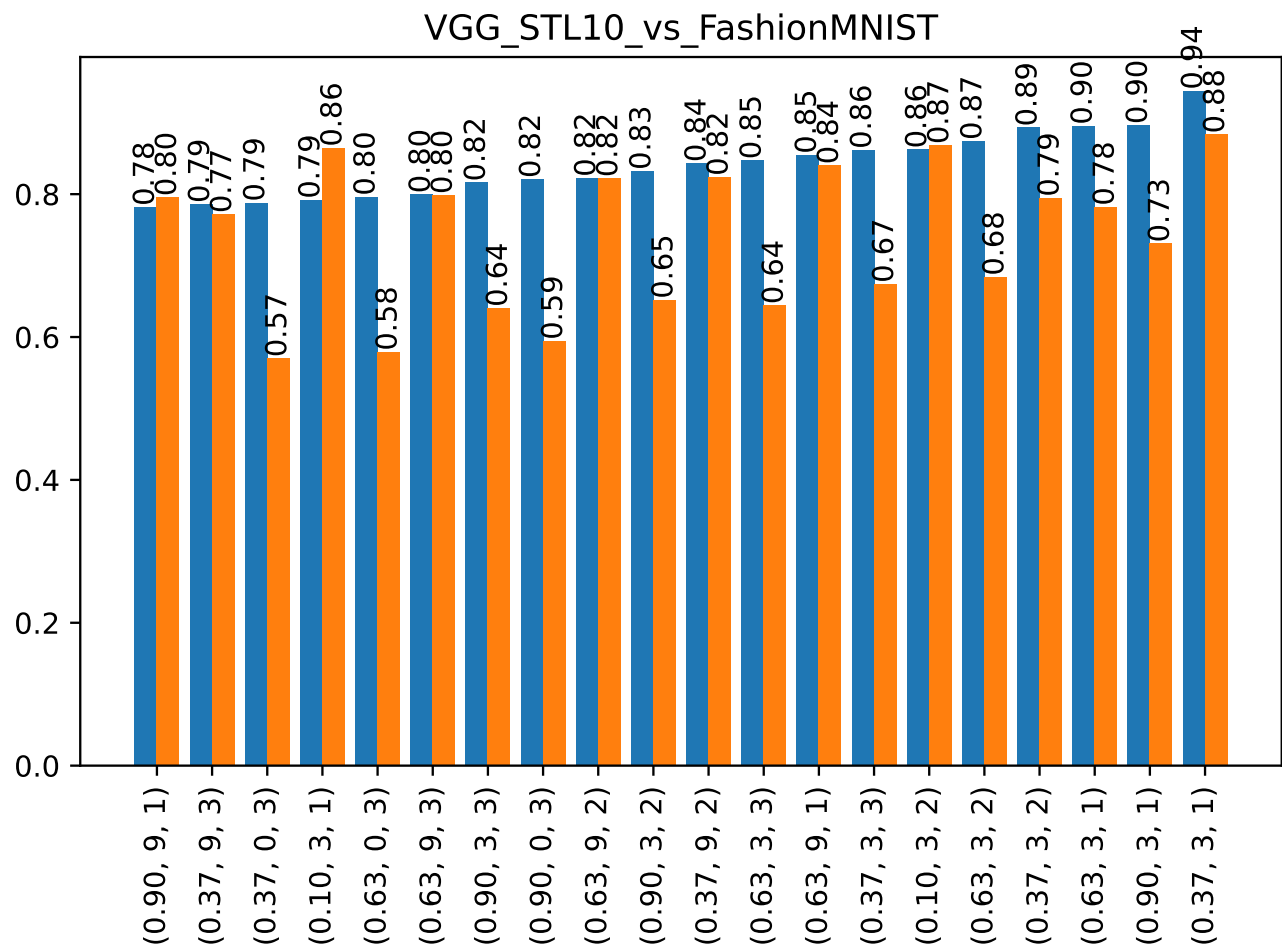


Figure 41: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

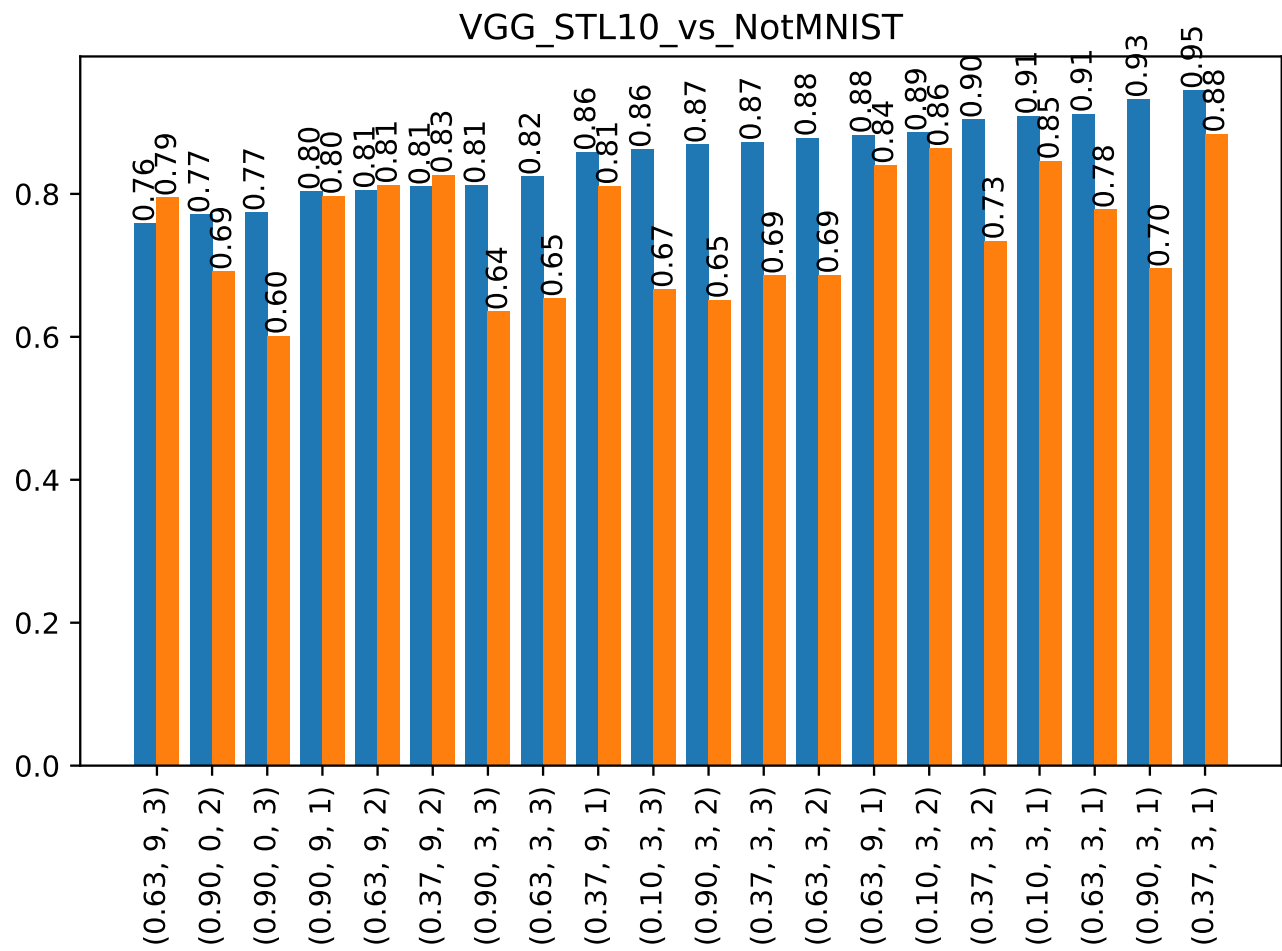


Figure 42: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

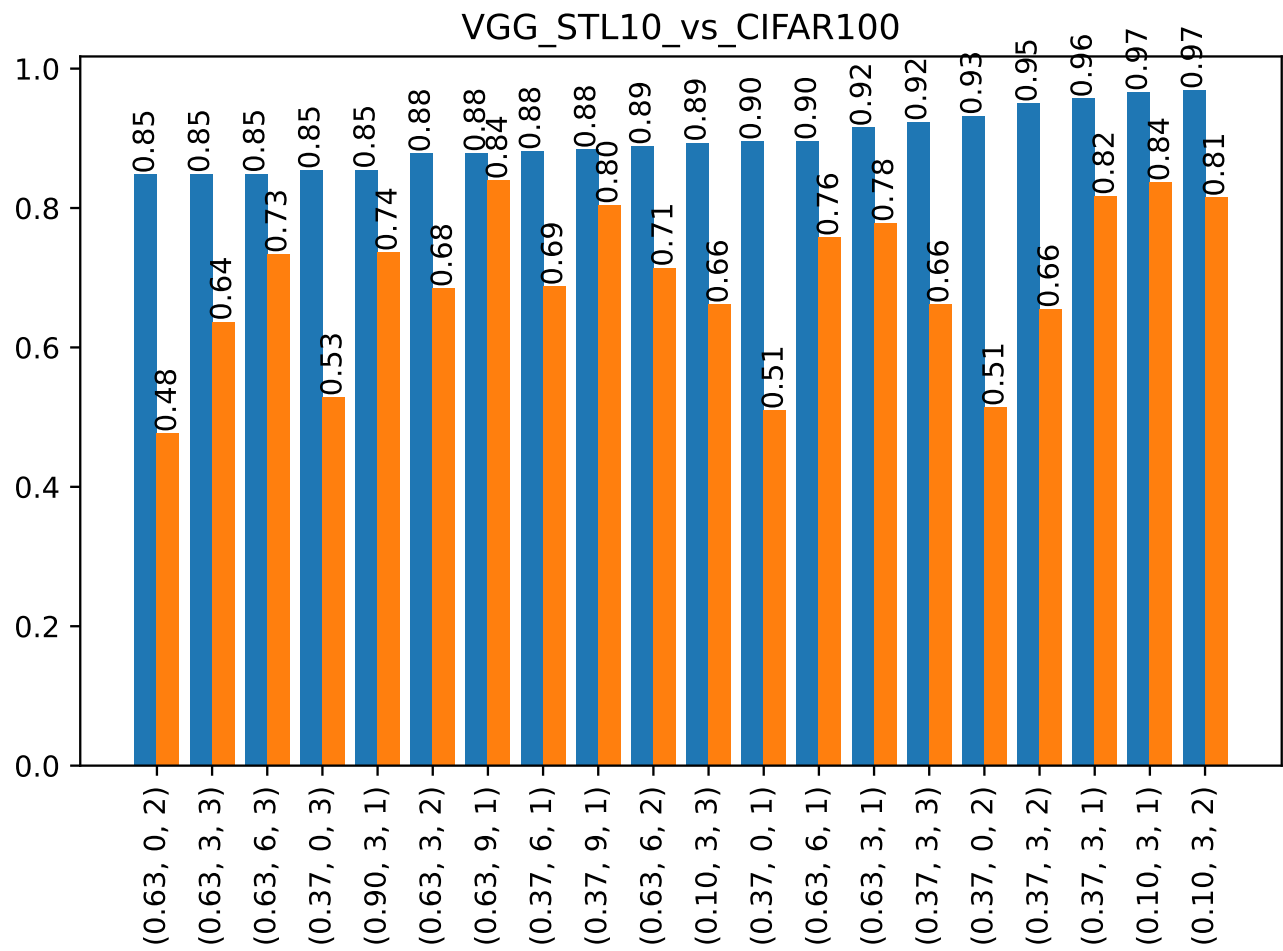


Figure 43: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

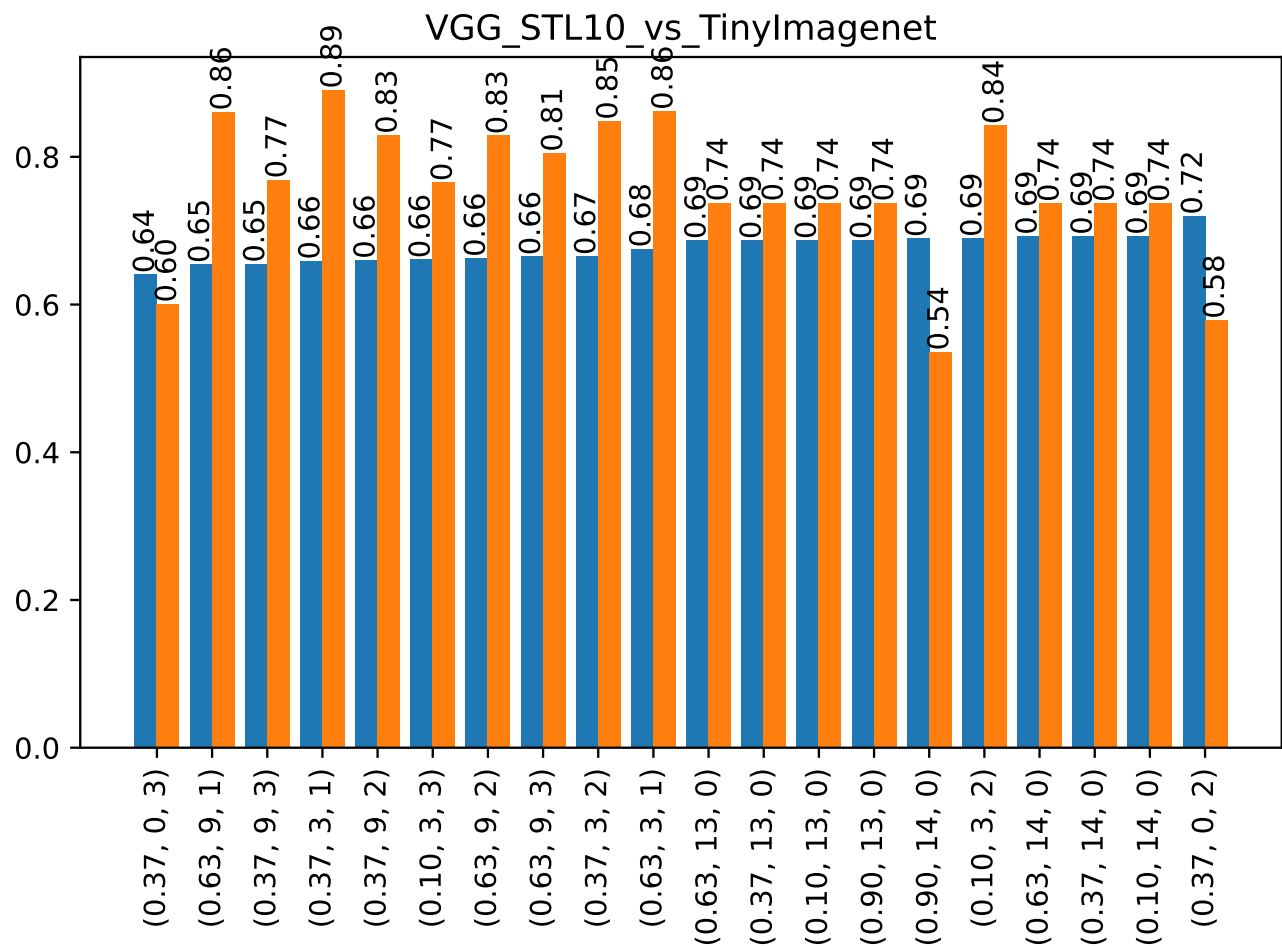


Figure 44: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

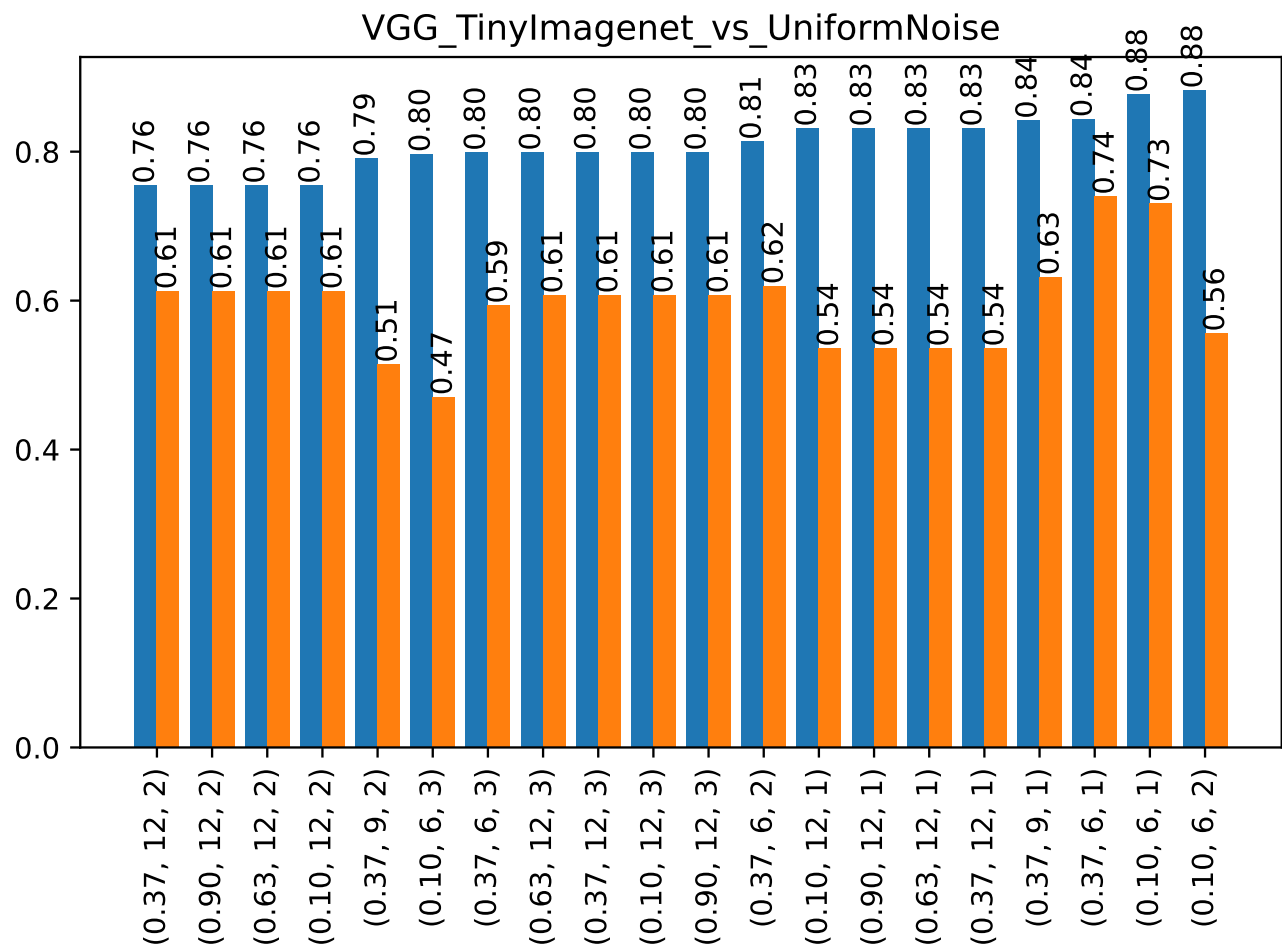


Figure 45: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaraneczowy).

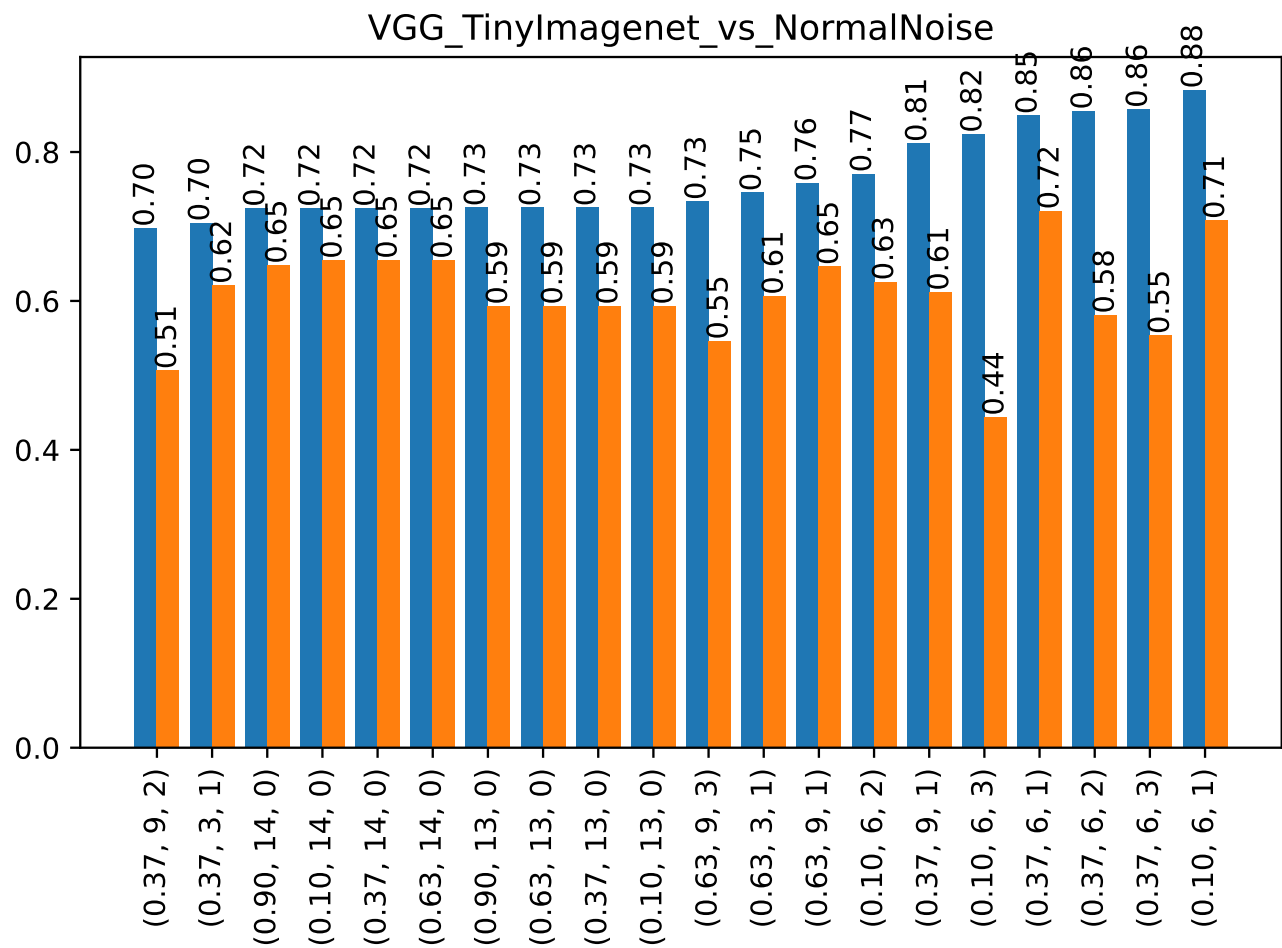


Figure 46: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

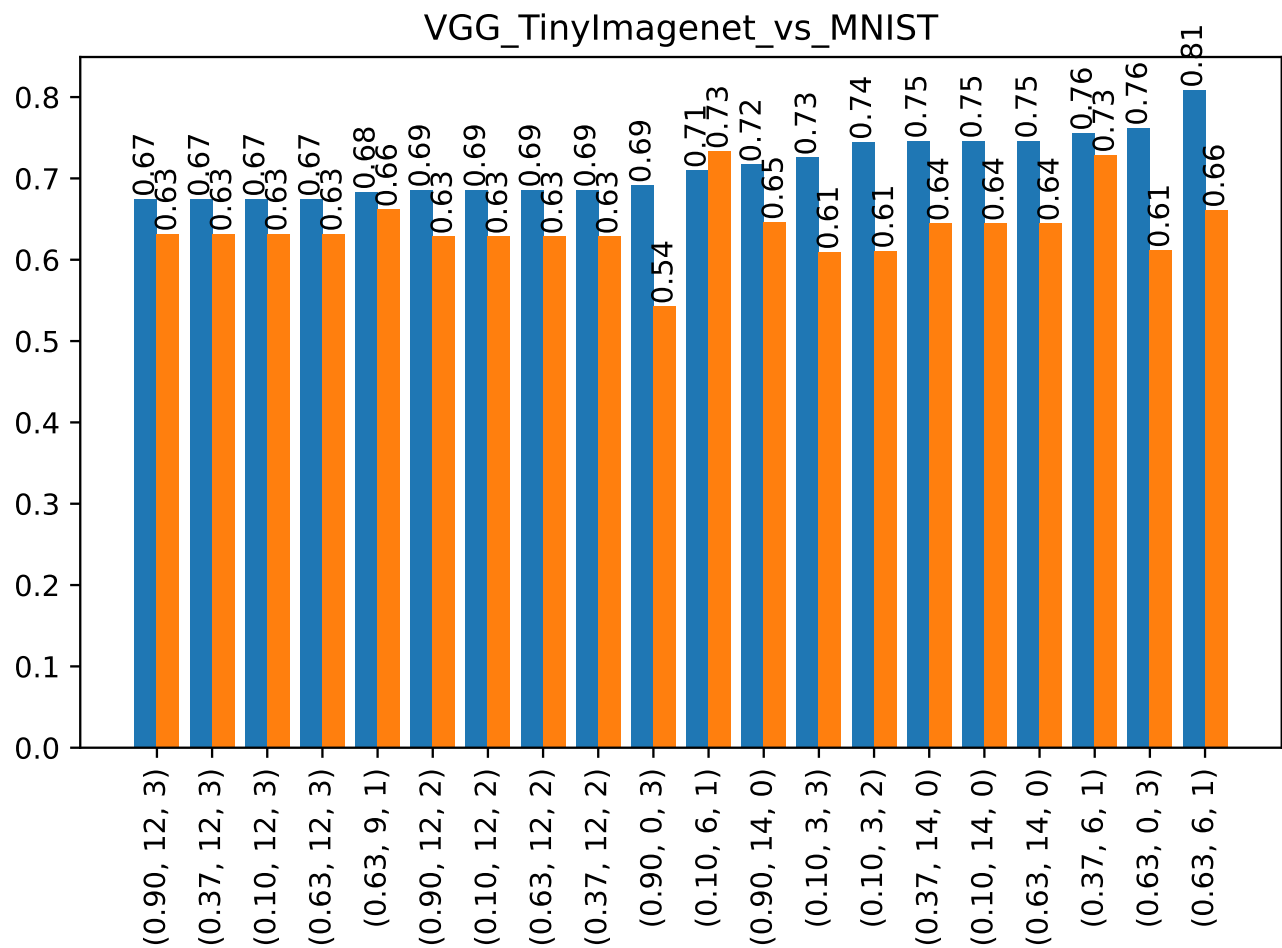


Figure 47: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

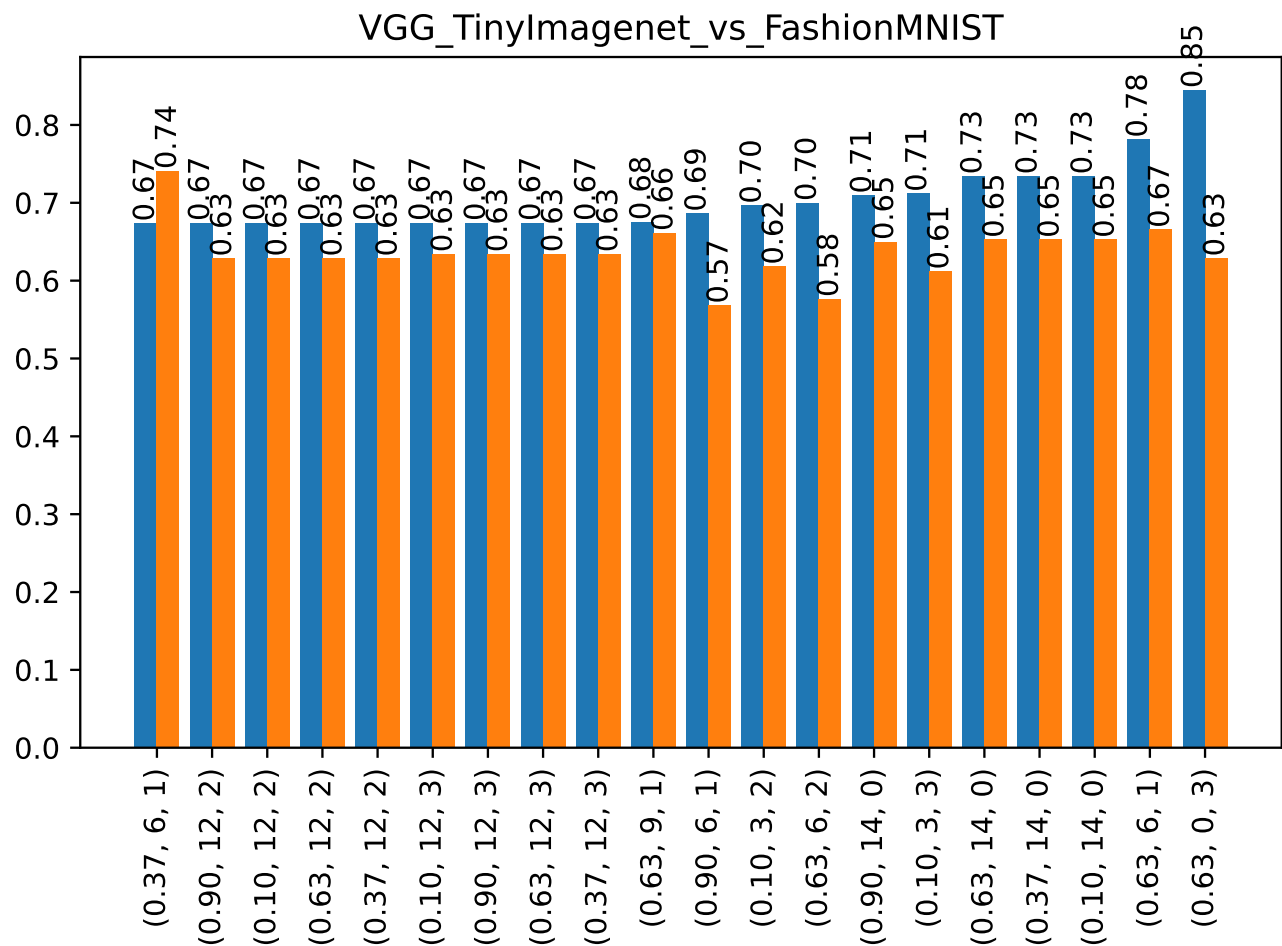


Figure 48: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

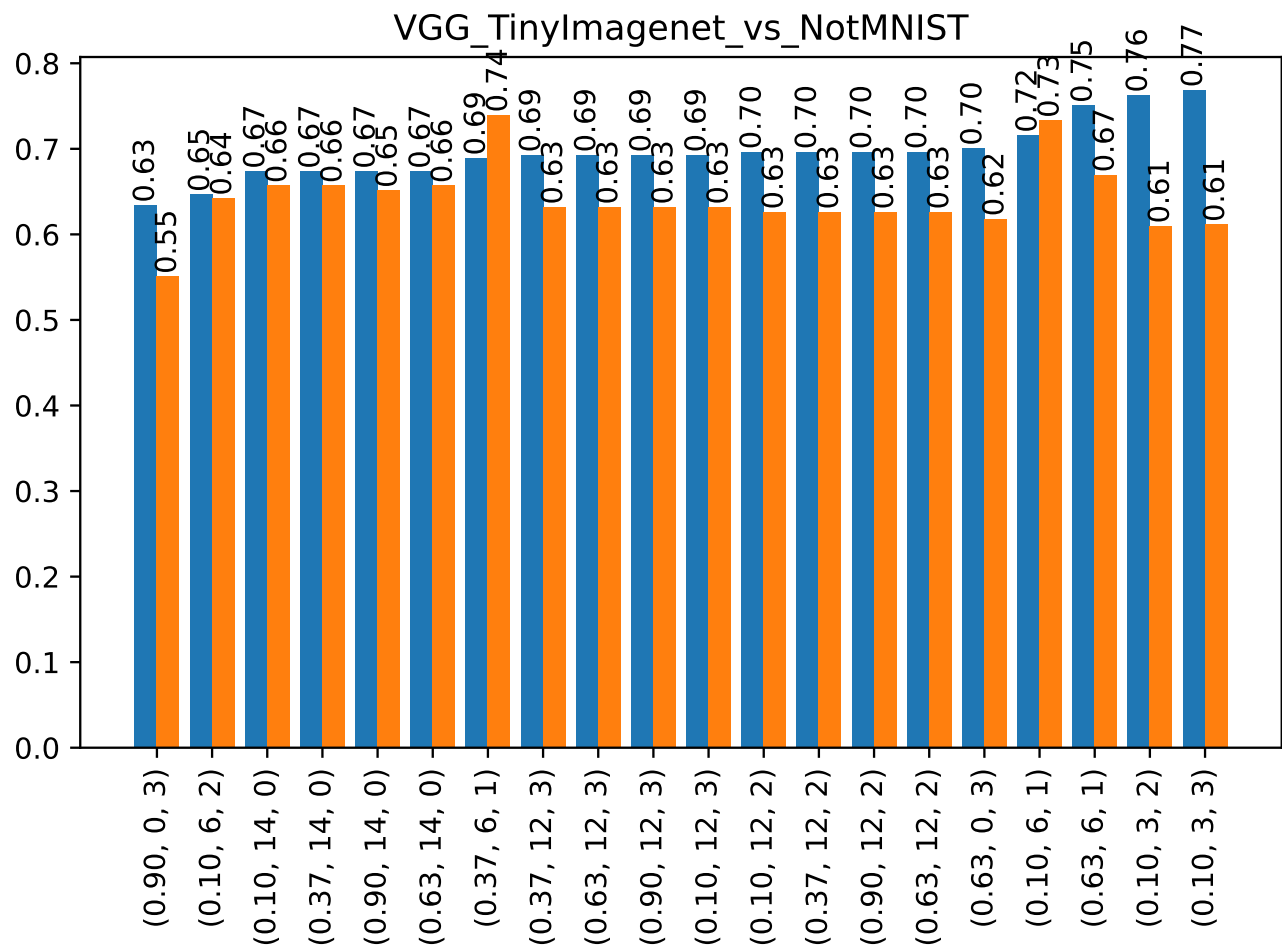


Figure 49: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomarańczowy).

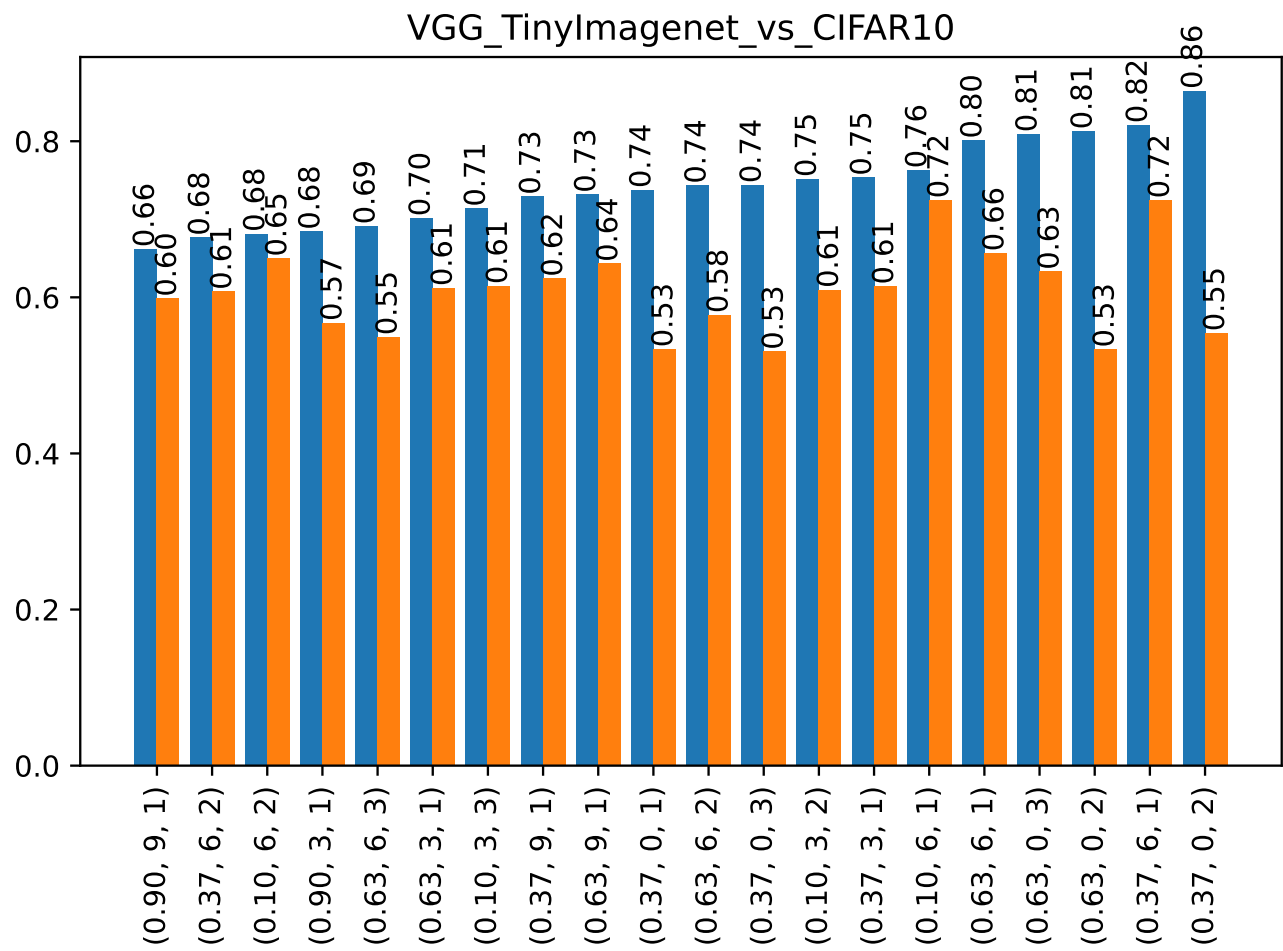


Figure 50: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

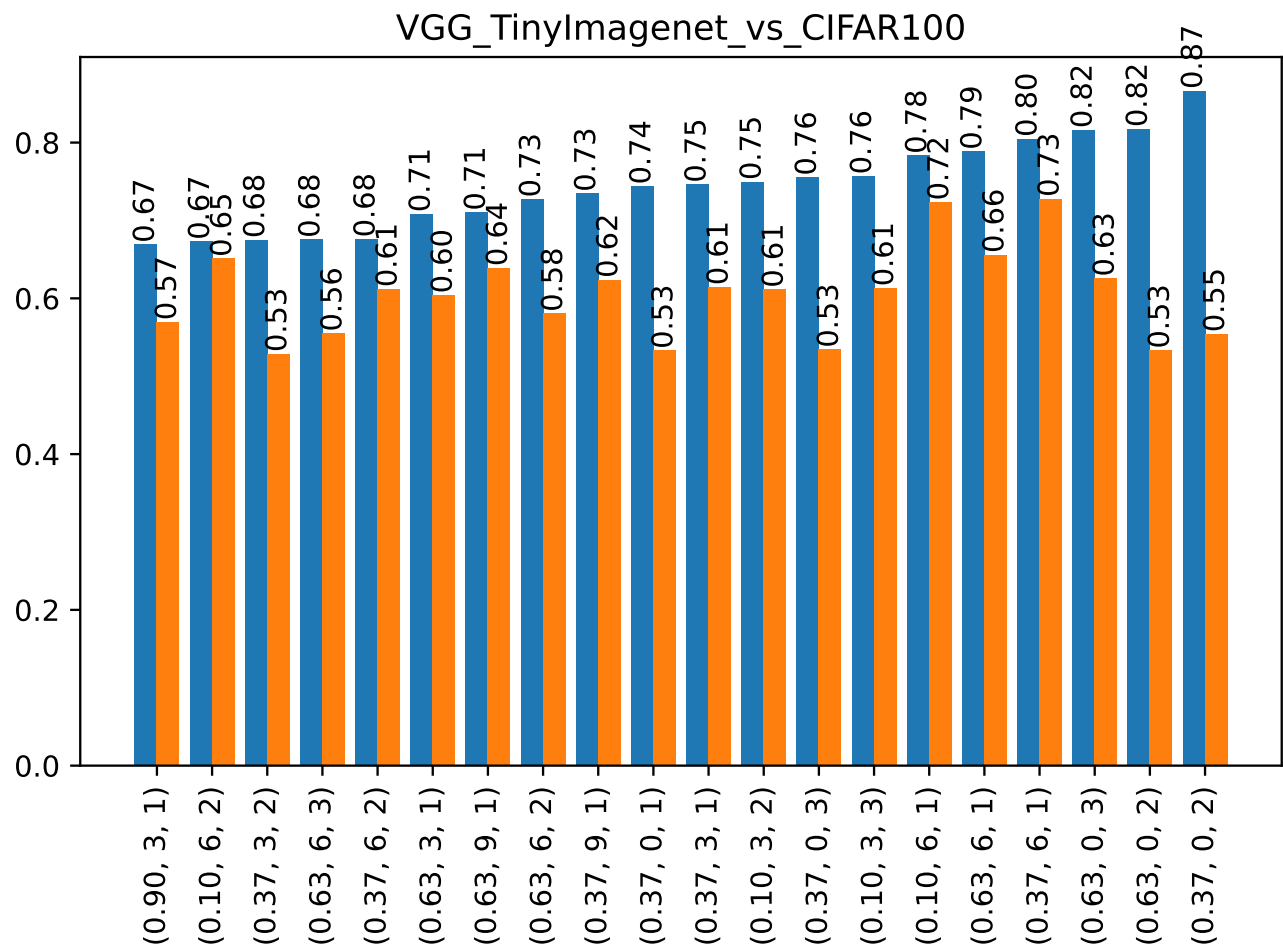


Figure 51: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaranczowy).

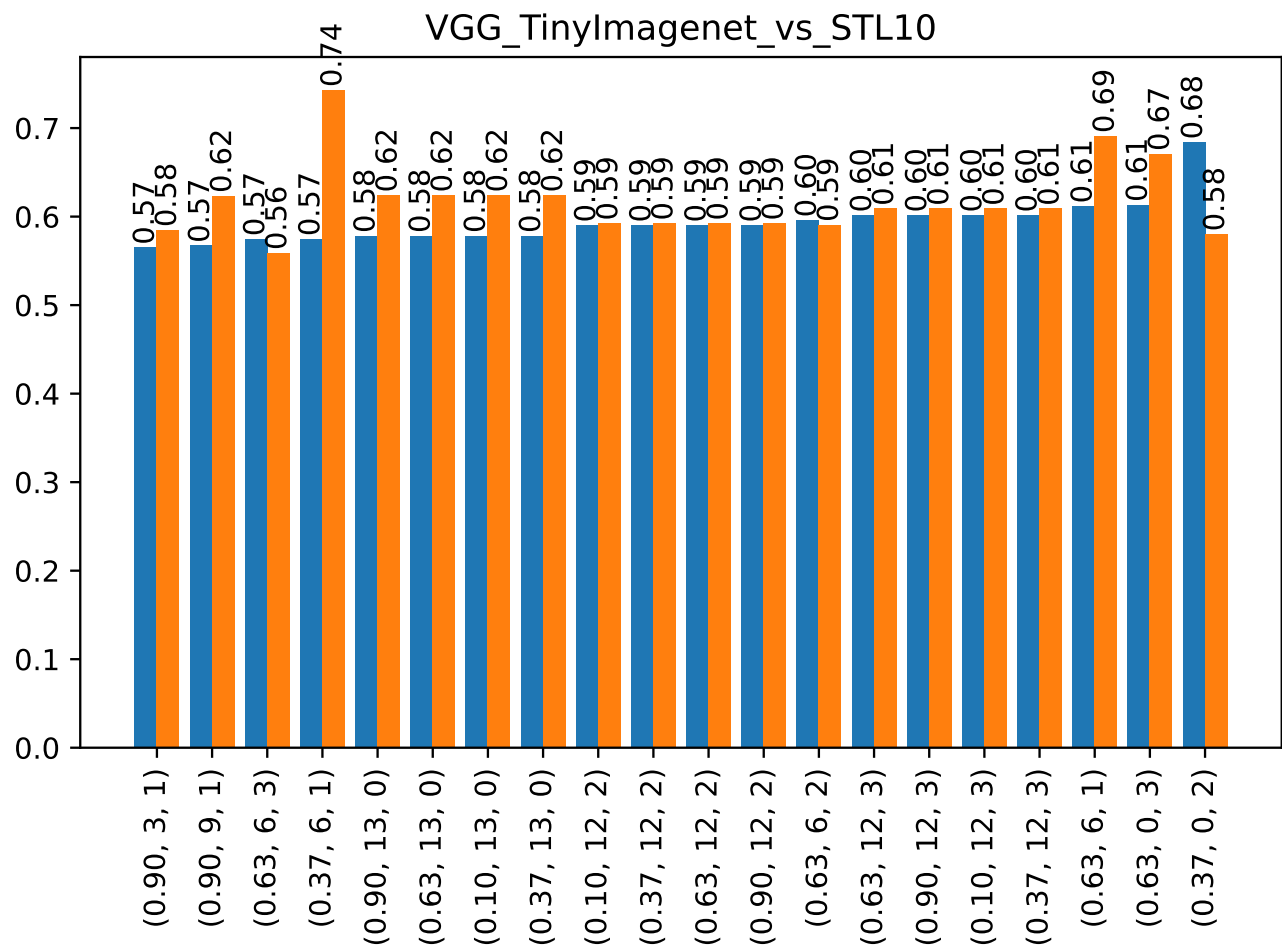


Figure 52: X - konfiguracje; Y - valid_accuracy (niebieski), test_accuracy (pomaraneczowy).

4 Wykresy dystansów wzorców aktywacji

VGGCIFAR100vsCIFAR10

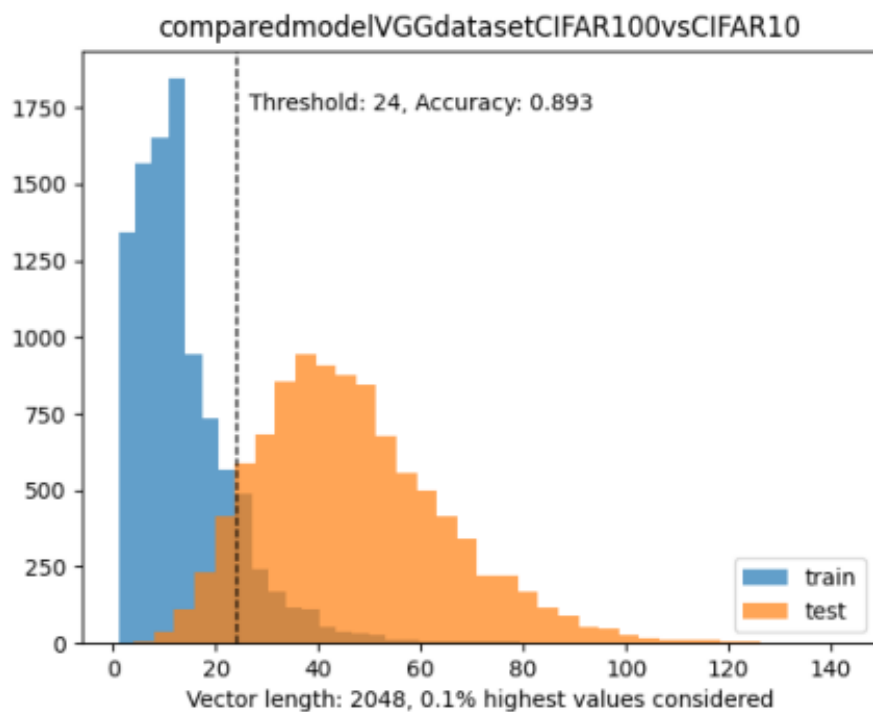


Figure 53: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsFashionMNIST

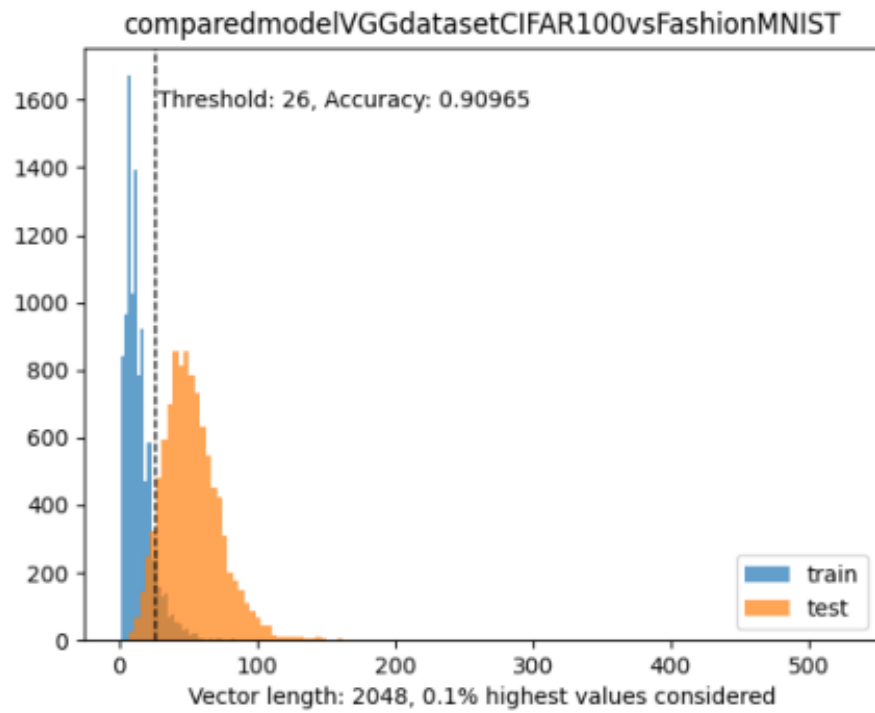


Figure 54: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsMNIST

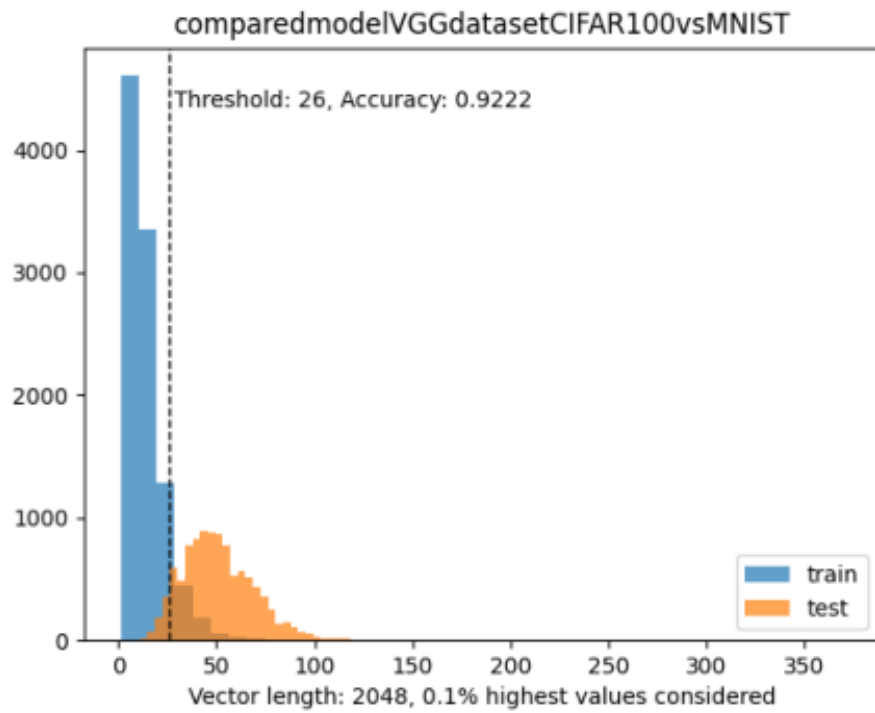


Figure 55: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsNormalNoise

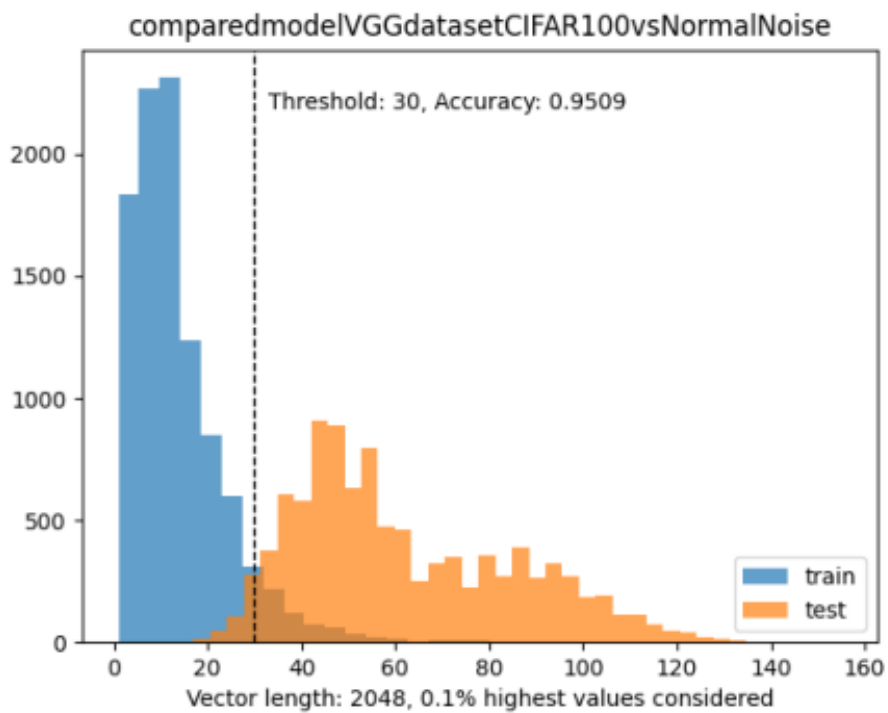


Figure 56: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsNotMNIST

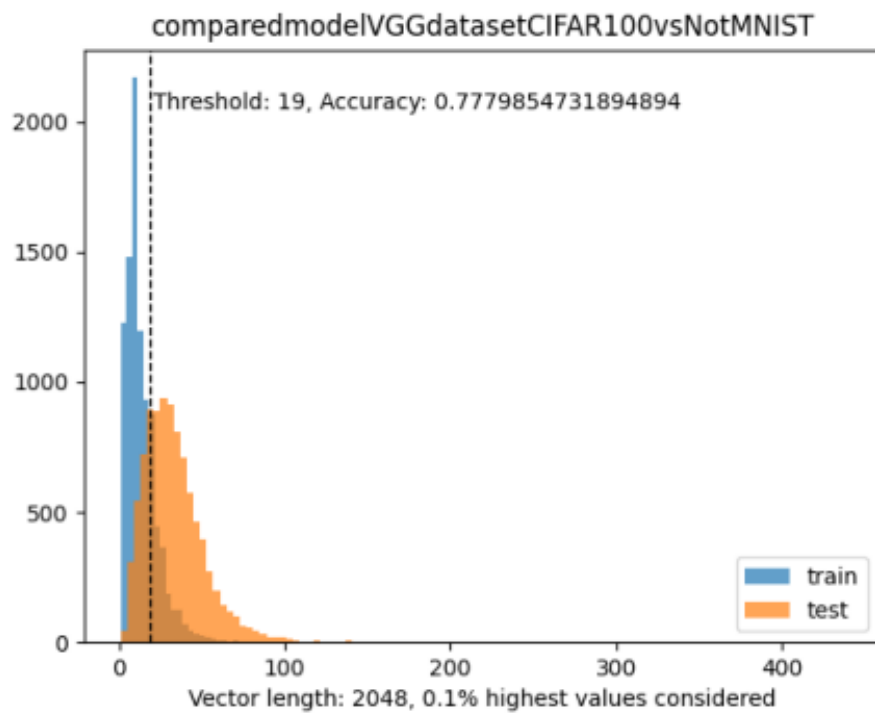


Figure 57: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsSTL10

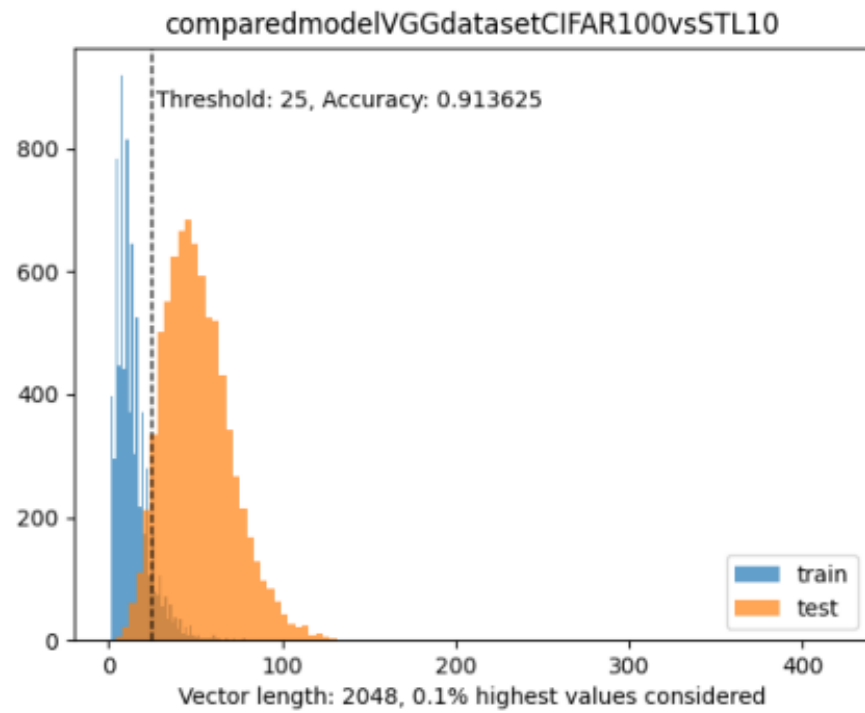


Figure 58: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsTinyImagenet

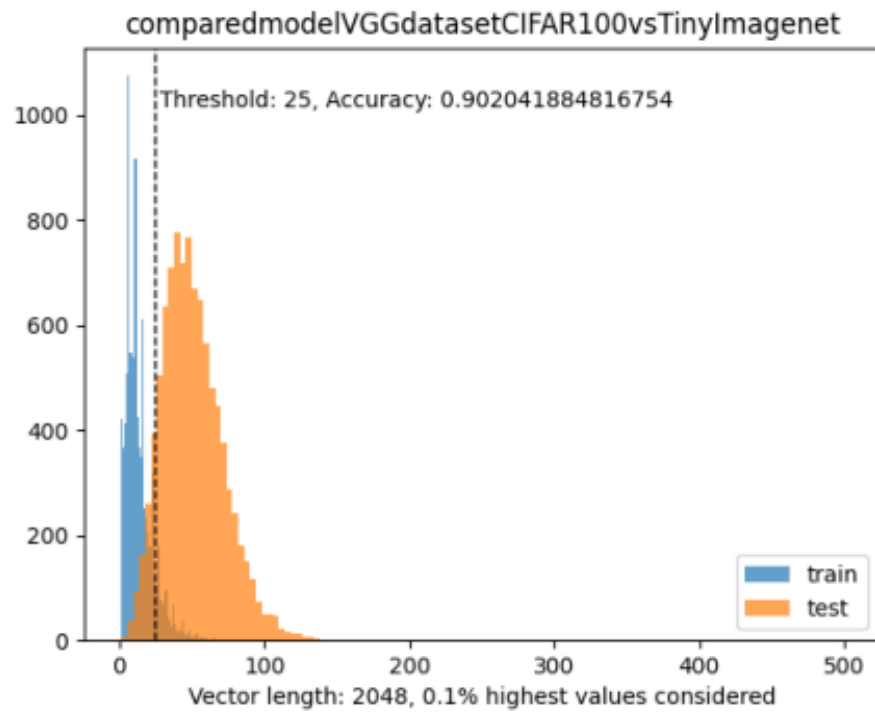


Figure 59: X - dystans; Y - liczba powtorzen.

VGGCIFAR100vsUniformNoise

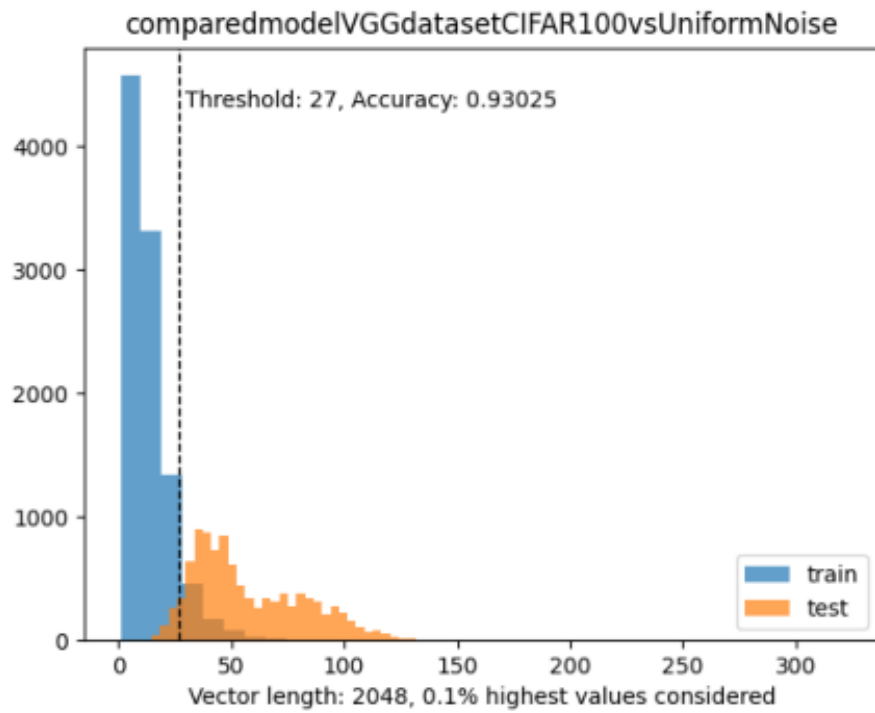


Figure 60: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsCIFAR100

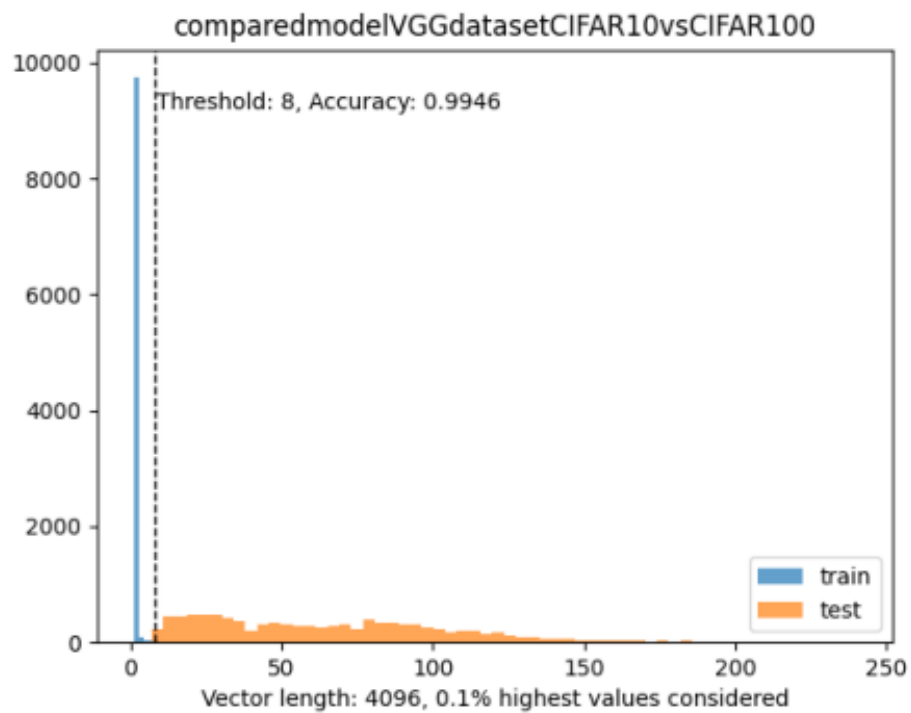


Figure 61: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsFashionMNIST

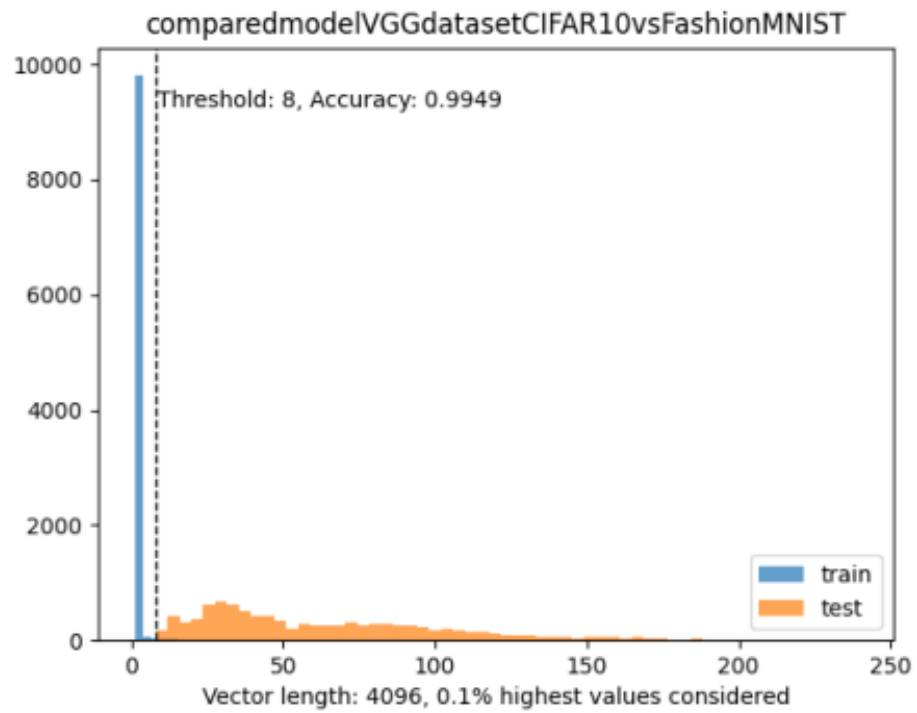


Figure 62: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsMNIST

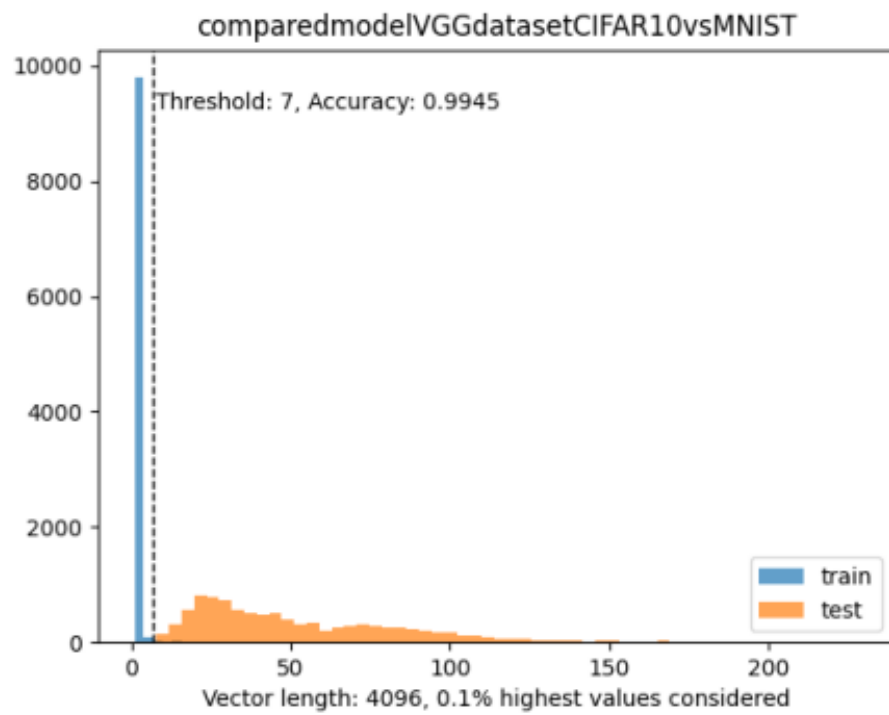


Figure 63: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsNormalNoise

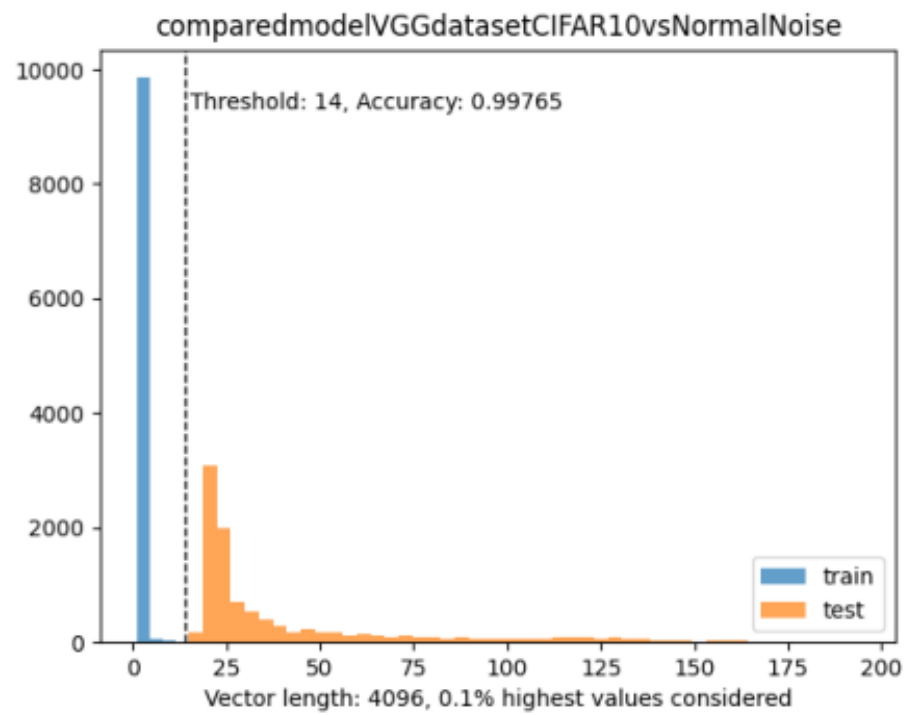


Figure 64: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsNotMNIST

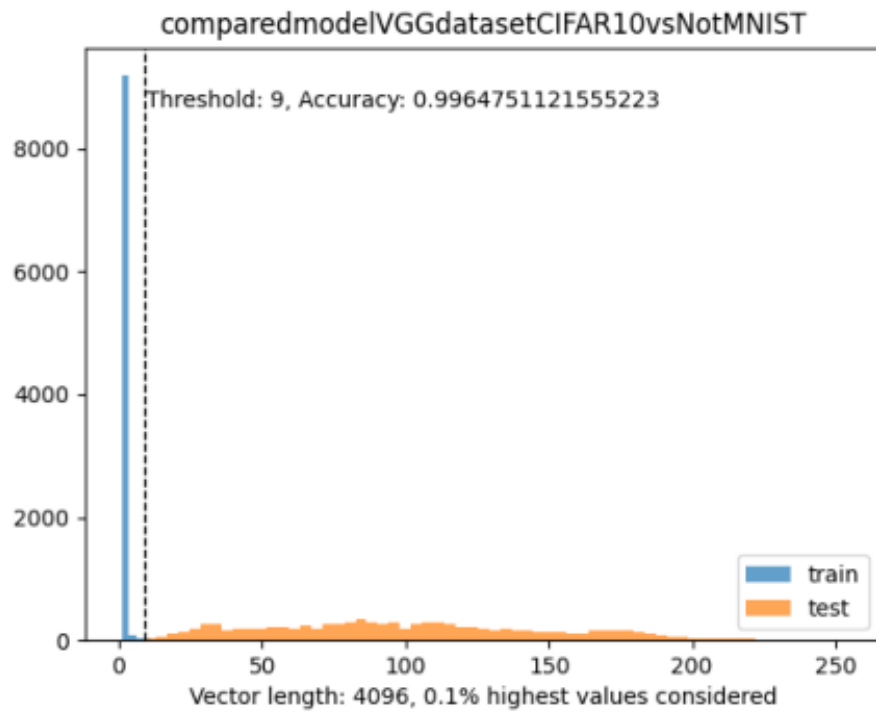


Figure 65: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsTinyImagenet

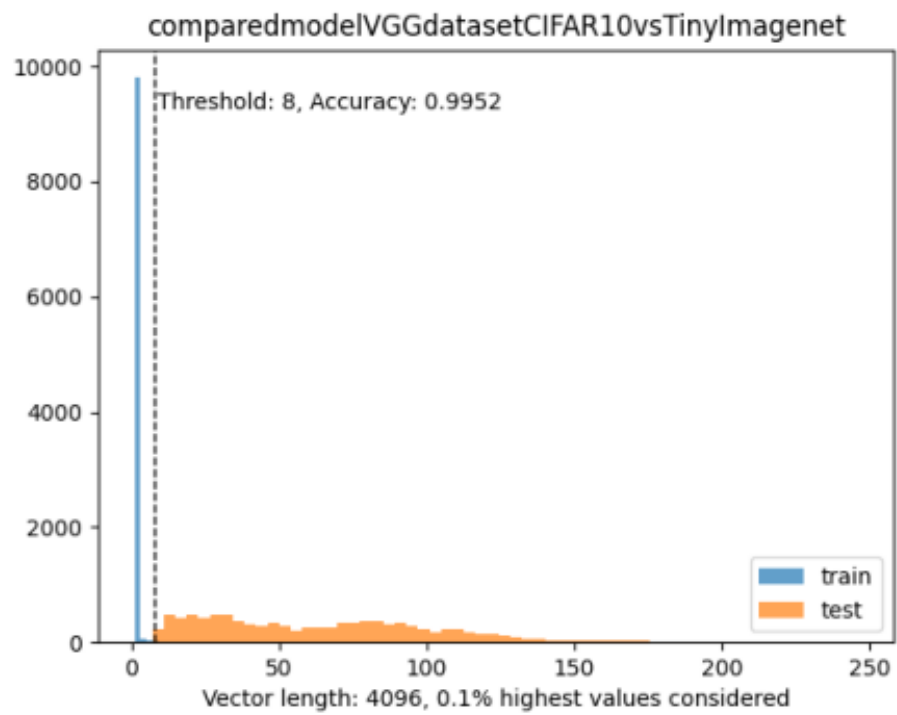


Figure 66: X - dystans; Y - liczba powtorzen.

VGGCIFAR10vsUniformNoise

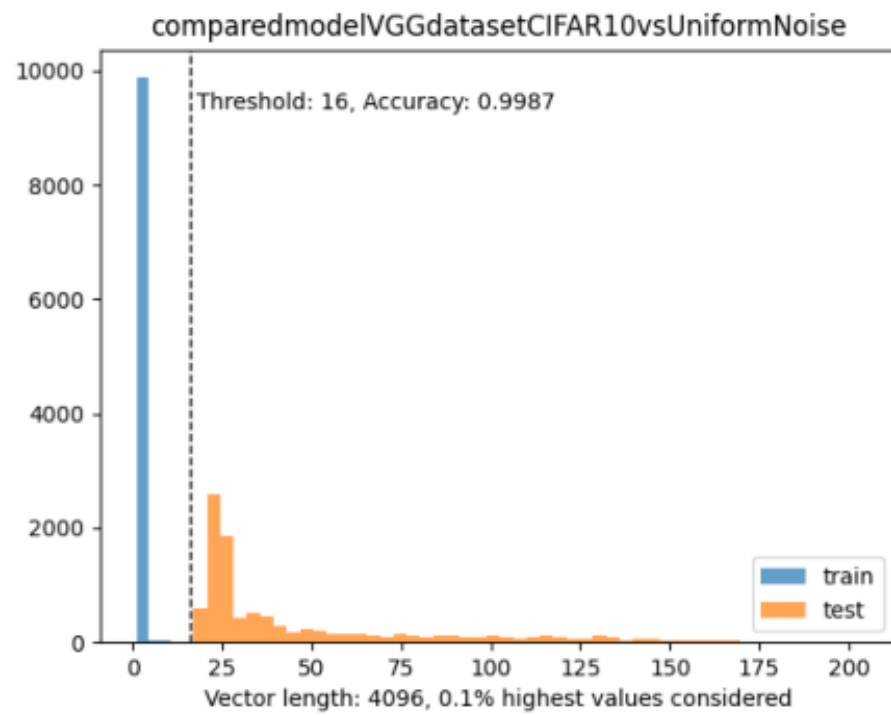


Figure 67: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsCIFAR10

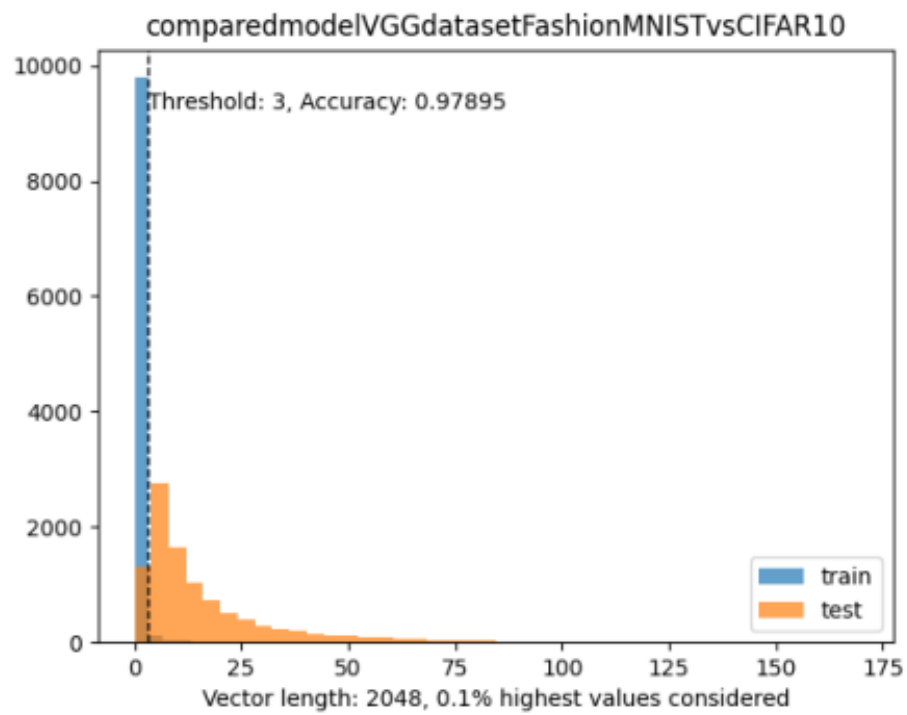


Figure 68: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsCIFAR100

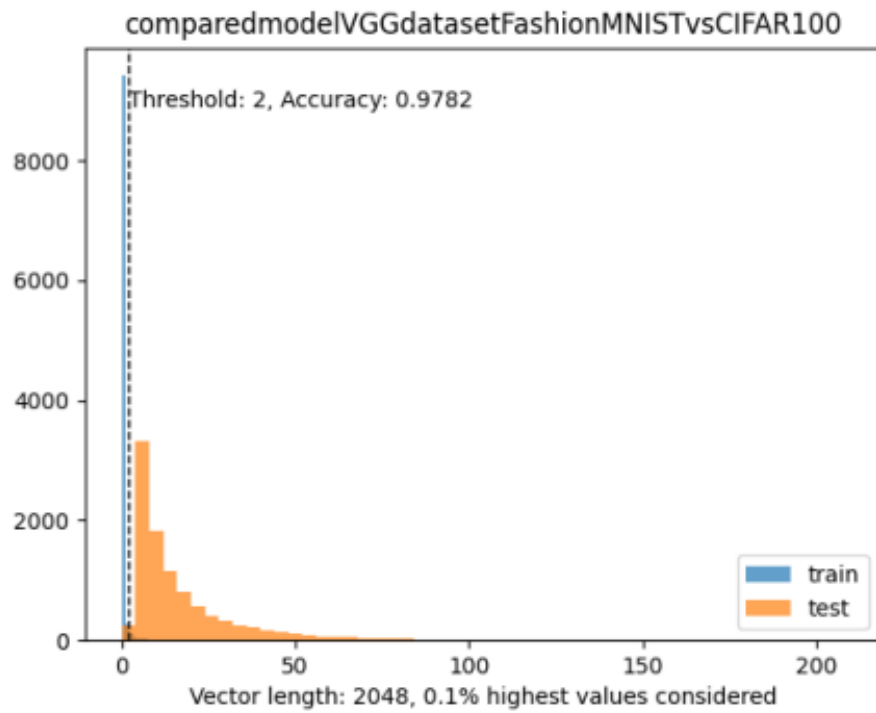


Figure 69: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsMNIST

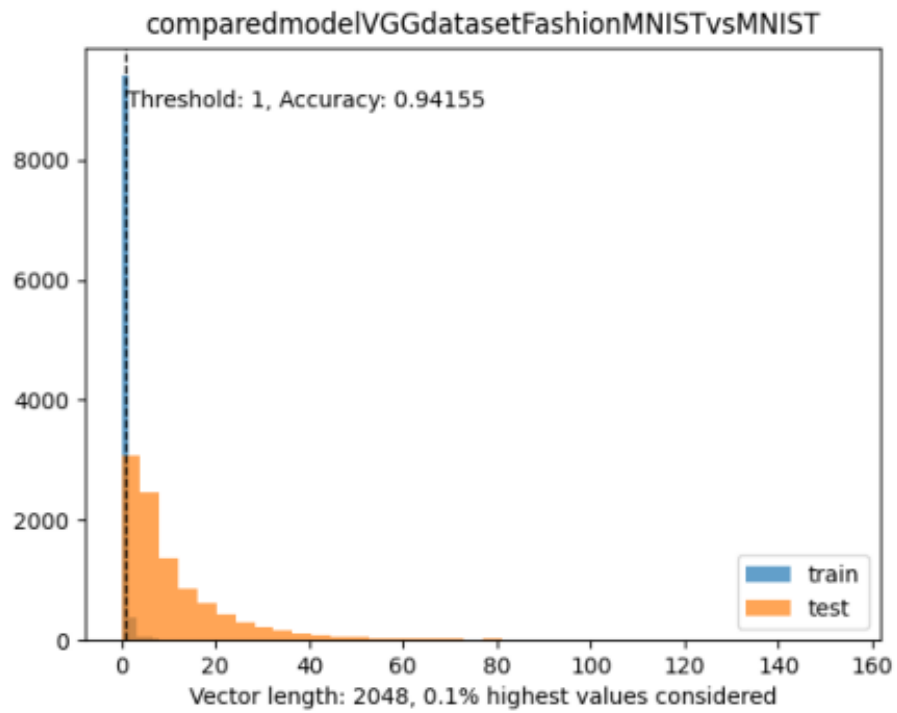


Figure 70: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsNormalNoise

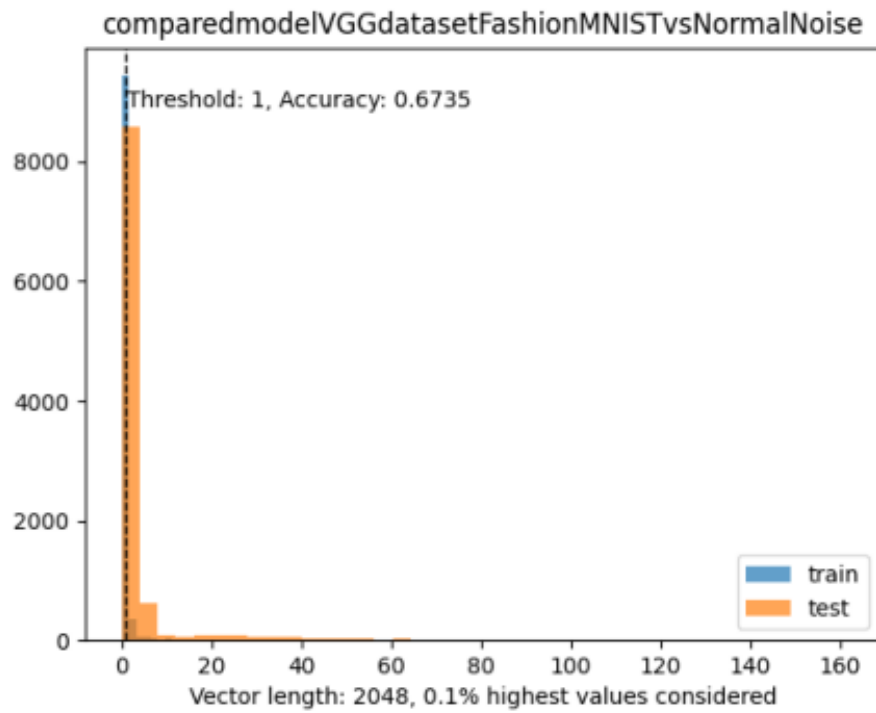


Figure 71: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsNotMNIST

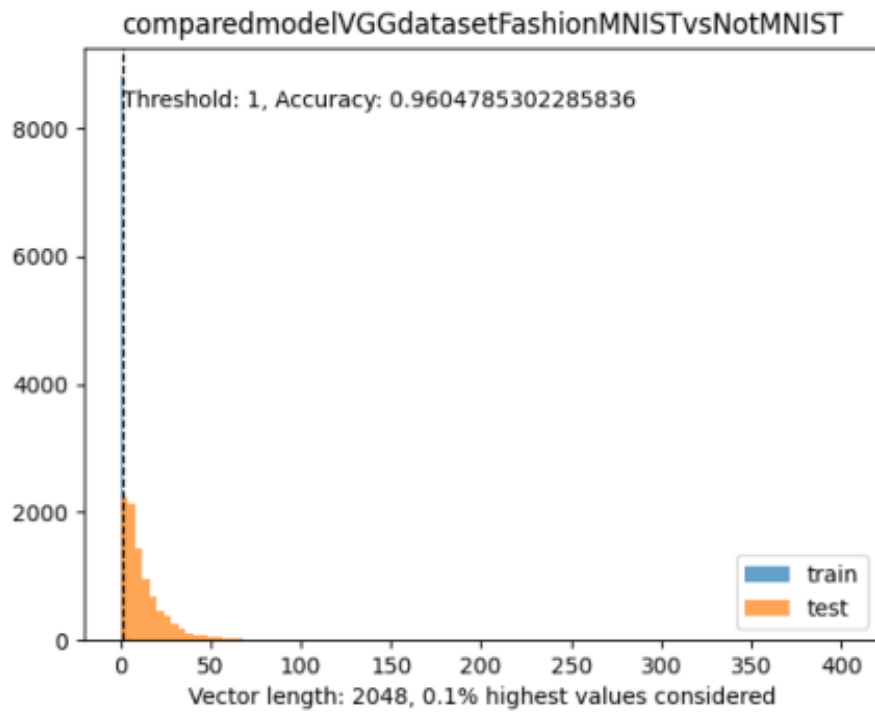


Figure 72: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsSTL10

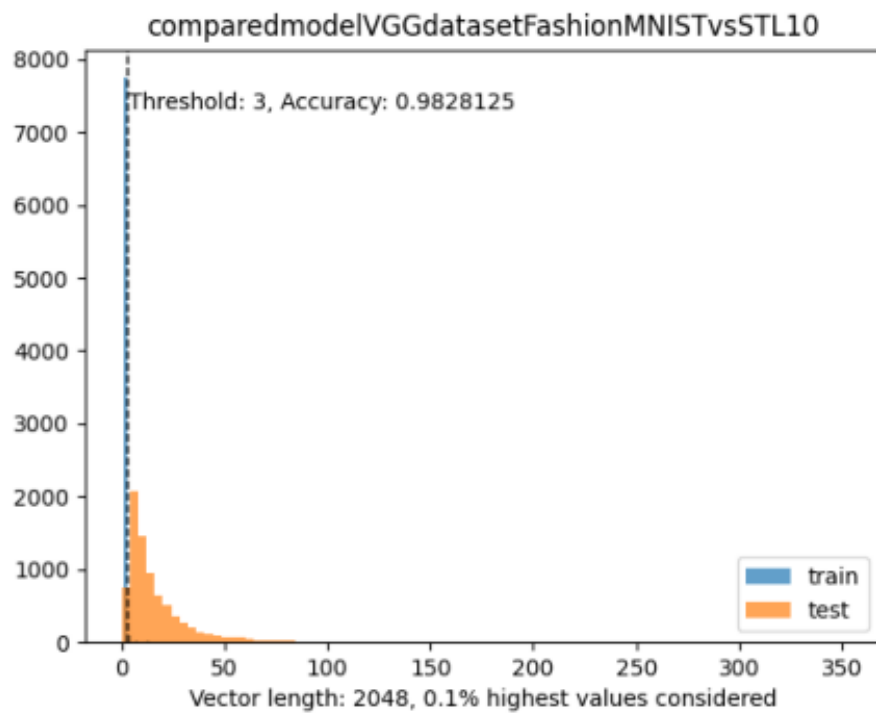


Figure 73: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsTinyImagenet

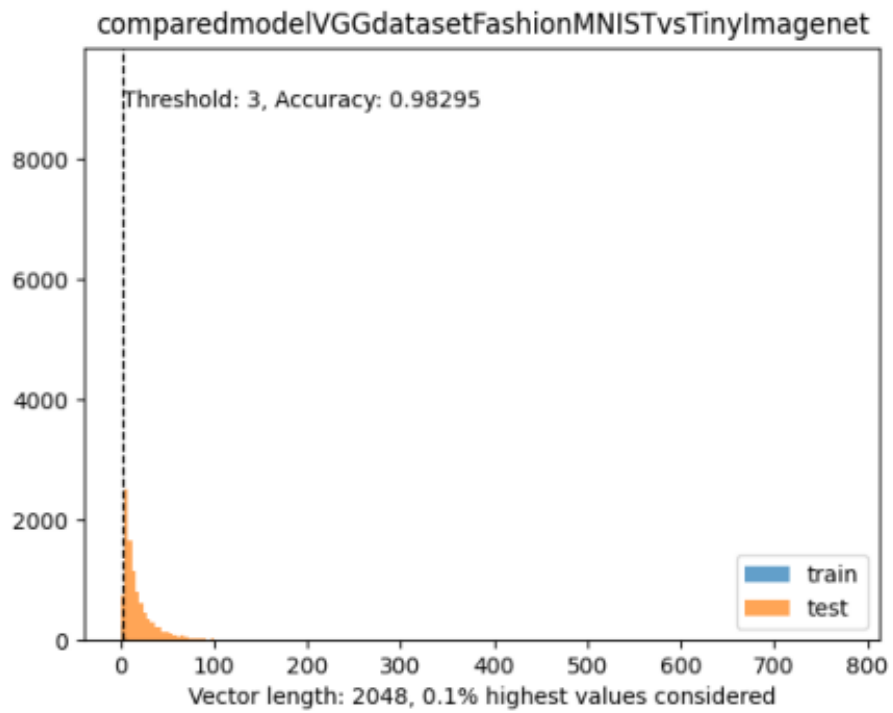


Figure 74: X - dystans; Y - liczba powtorzen.

VGGFashionMNISTvsUniformNoise

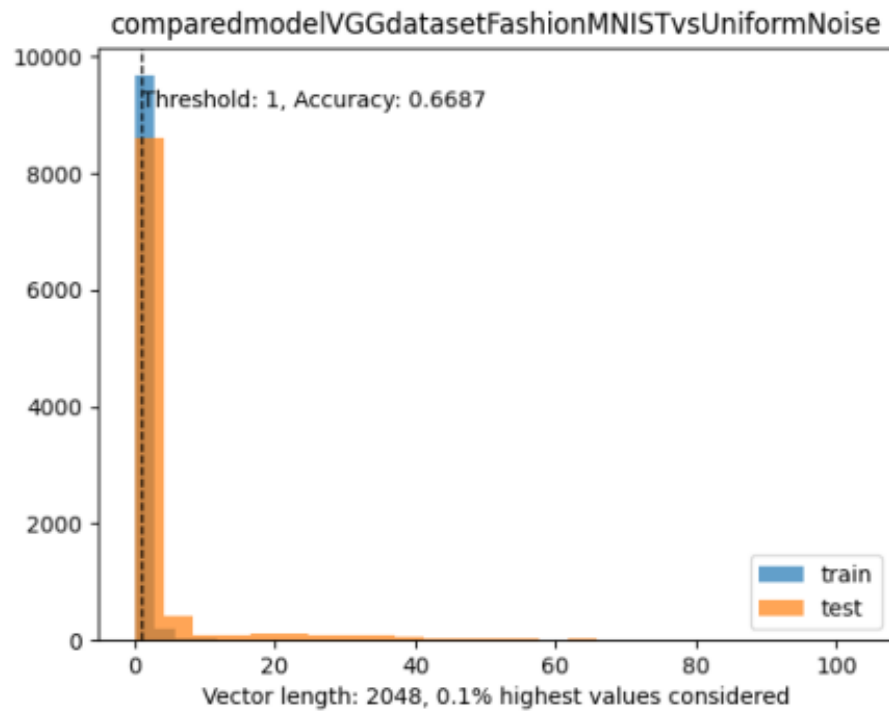


Figure 75: X - dystans; Y - liczba powtorzen.

VGGMNISTvsCIFAR10

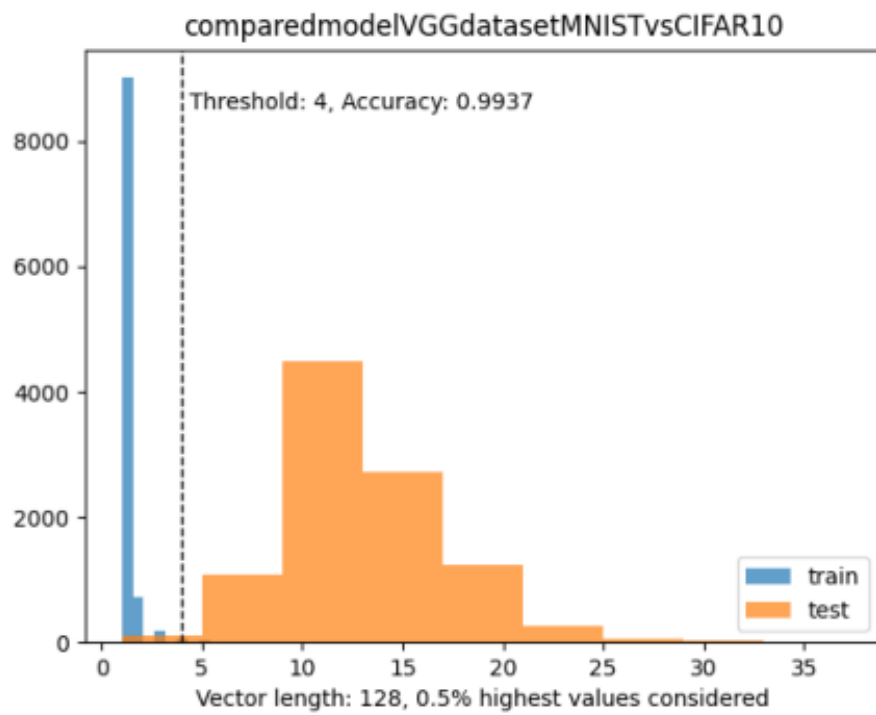


Figure 76: X - dystans; Y - liczba powtorzen.

VGGMNISTvsCIFAR100

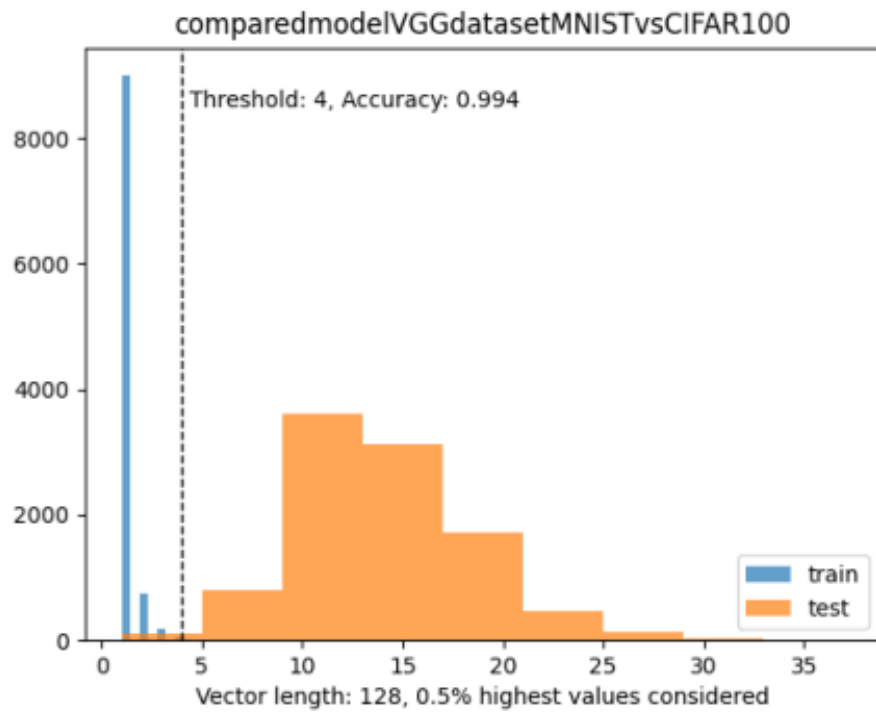


Figure 77: X - dystans; Y - liczba powtorzen.

VGGMNISTvsFashionMNIST

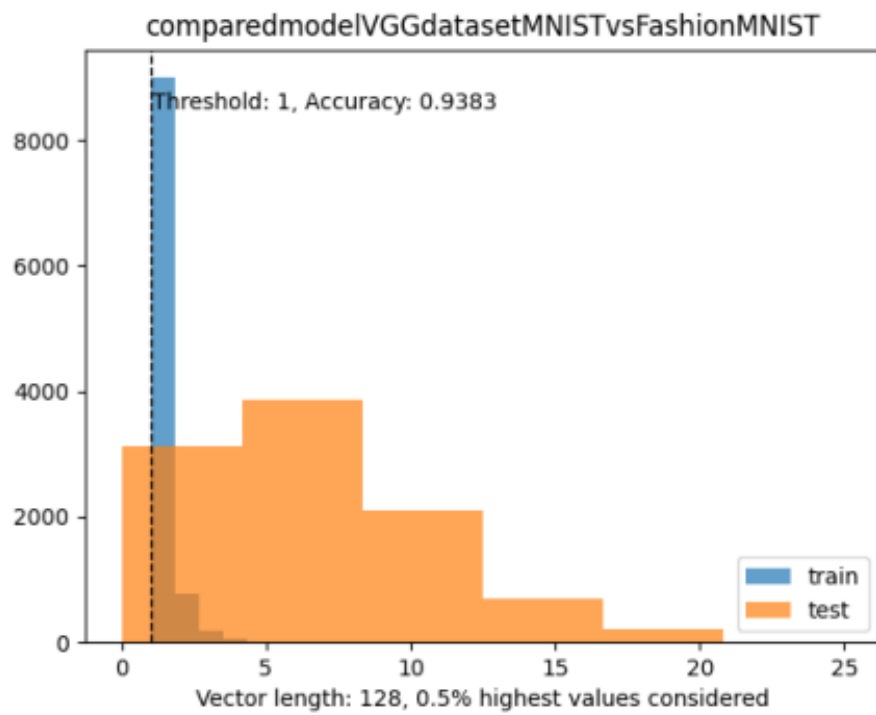


Figure 78: X - dystans; Y - liczba powtorzen.

VGGMNISTvsNormalNoise

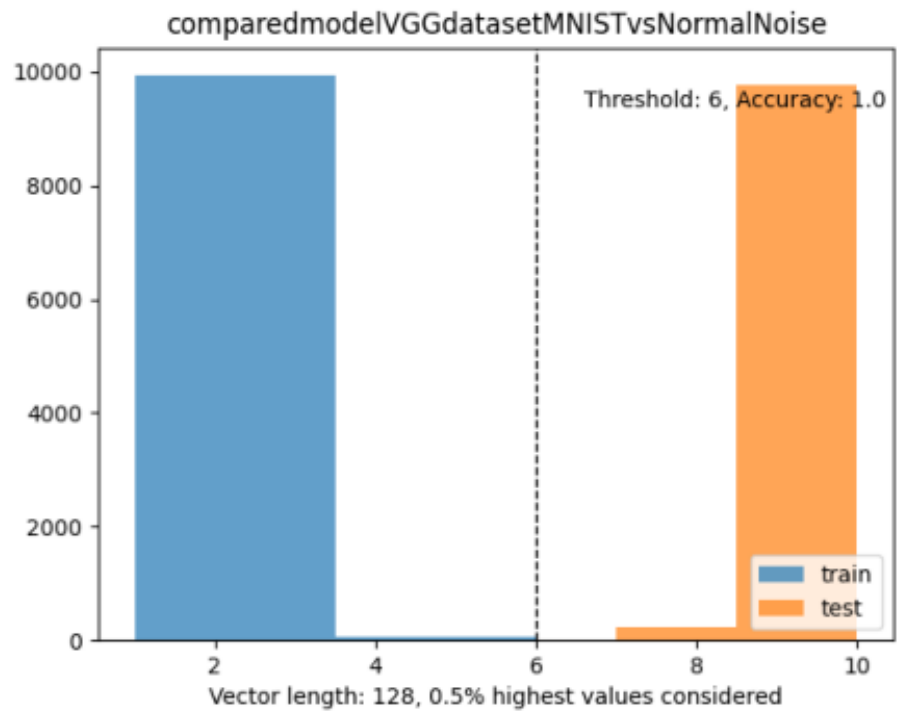


Figure 79: X - dystans; Y - liczba powtorzen.

VGGMNISTvsNotMNIST

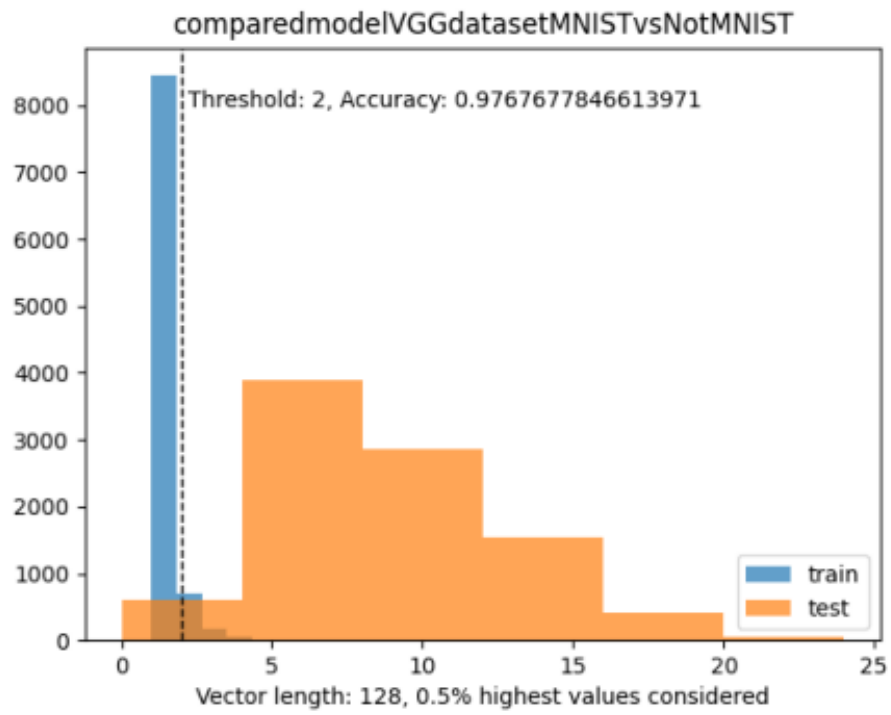


Figure 80: X - dystans; Y - liczba powtorzen.

VGGMNISTvsSTL10

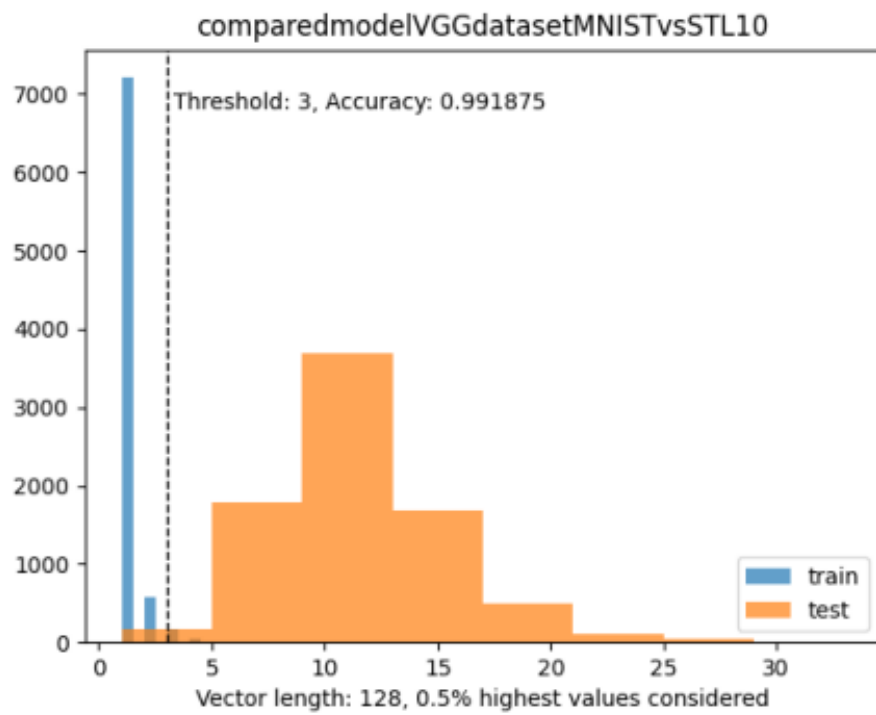


Figure 81: X - dystans; Y - liczba powtorzen.

VGGMNISTvsTinyImagenet

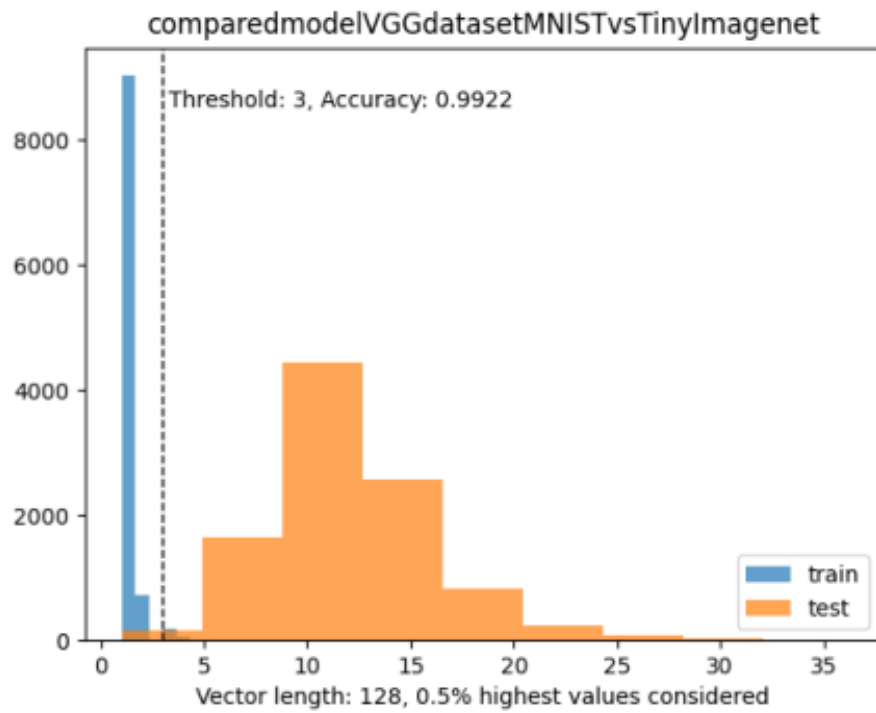


Figure 82: X - dystans; Y - liczba powtorzen.

VGGMNISTvsUniformNoise

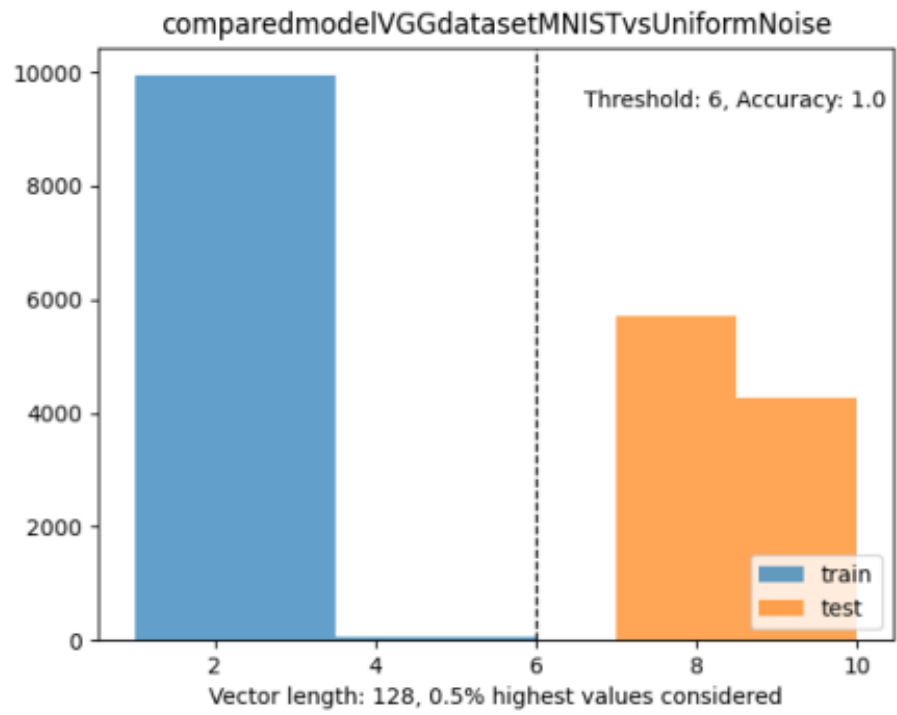


Figure 83: X - dystans; Y - liczba powtorzen.

VGGSTL10vsCIFAR100

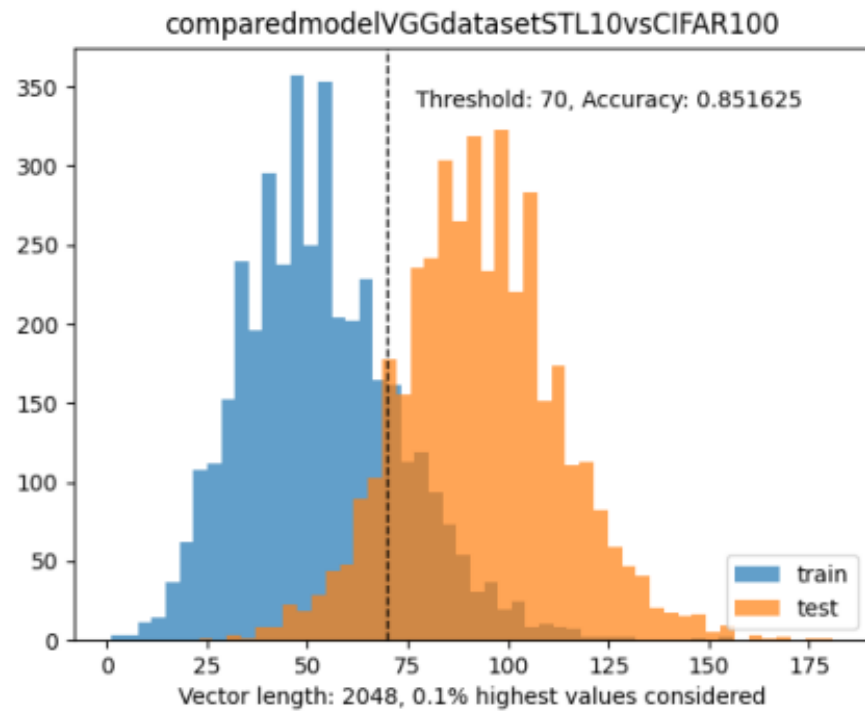


Figure 84: X - dystans; Y - liczba powtorzen.

VGGSTL10vsFashionMNIST

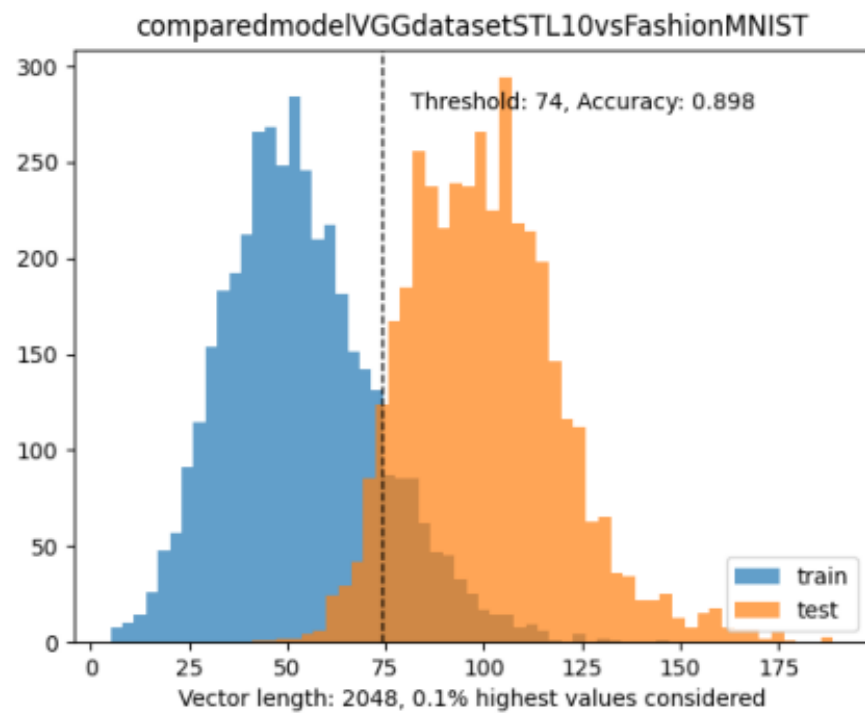


Figure 85: X - dystans; Y - liczba powtorzen.

VGGSTL10vsMNIST

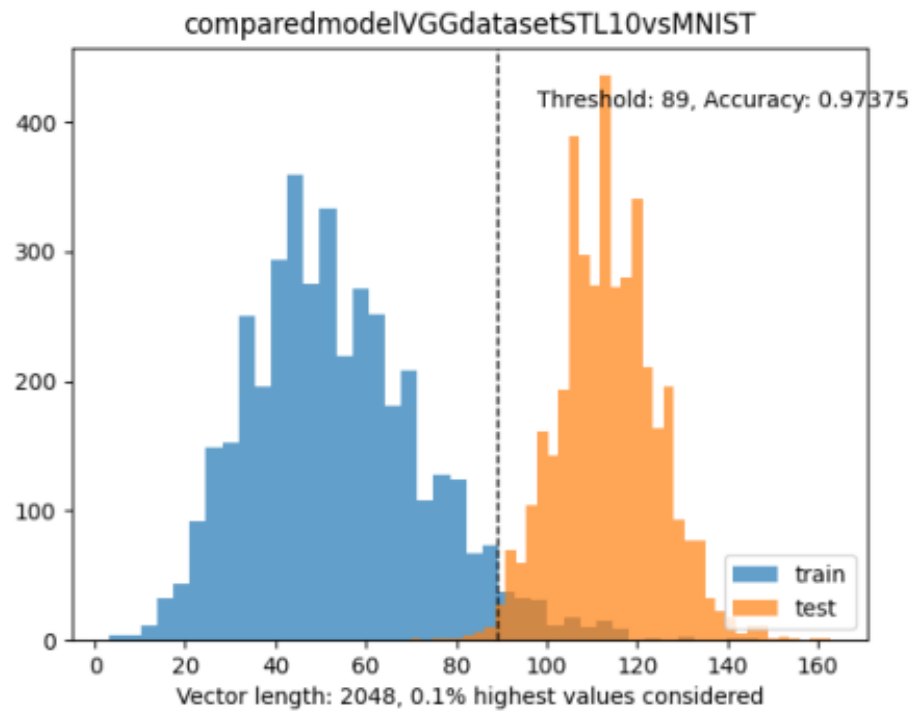


Figure 86: X - dystans; Y - liczba powtorzen.

VGGSTL10vsNormalNoise

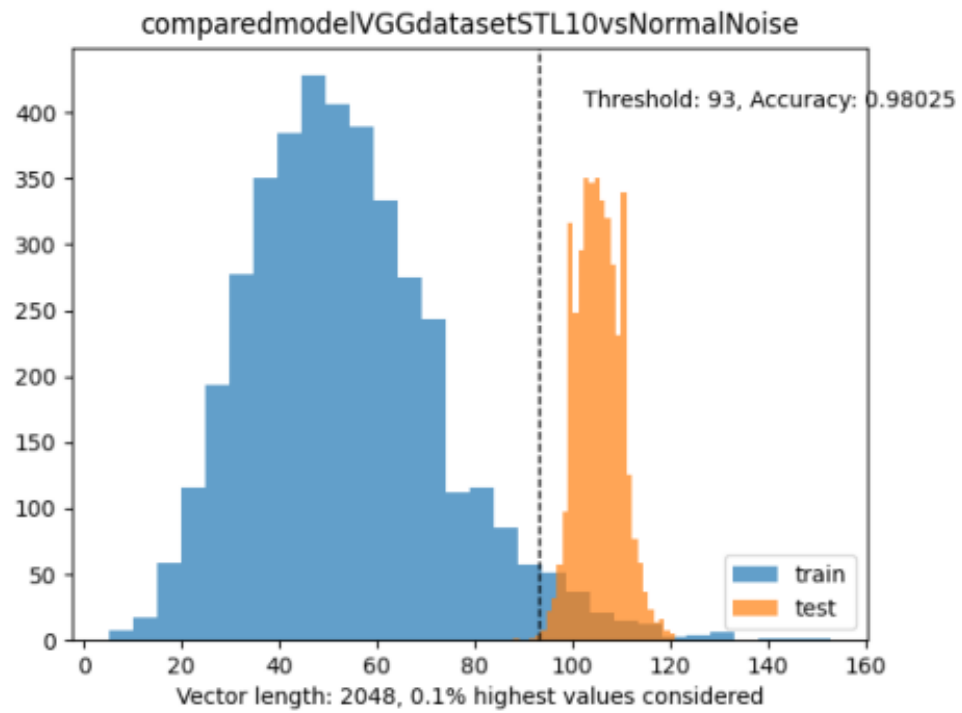


Figure 87: X - dystans; Y - liczba powtorzen.

VGGSTL10vsNotMNIST

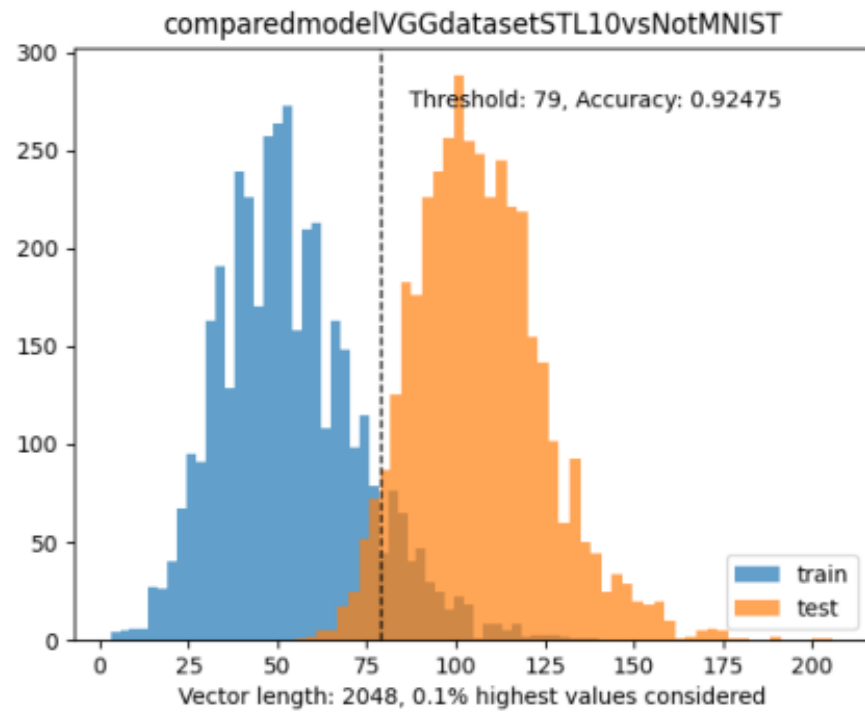


Figure 88: X - dystans; Y - liczba powtorzen.

VGGSTL10vsTinyImagenet

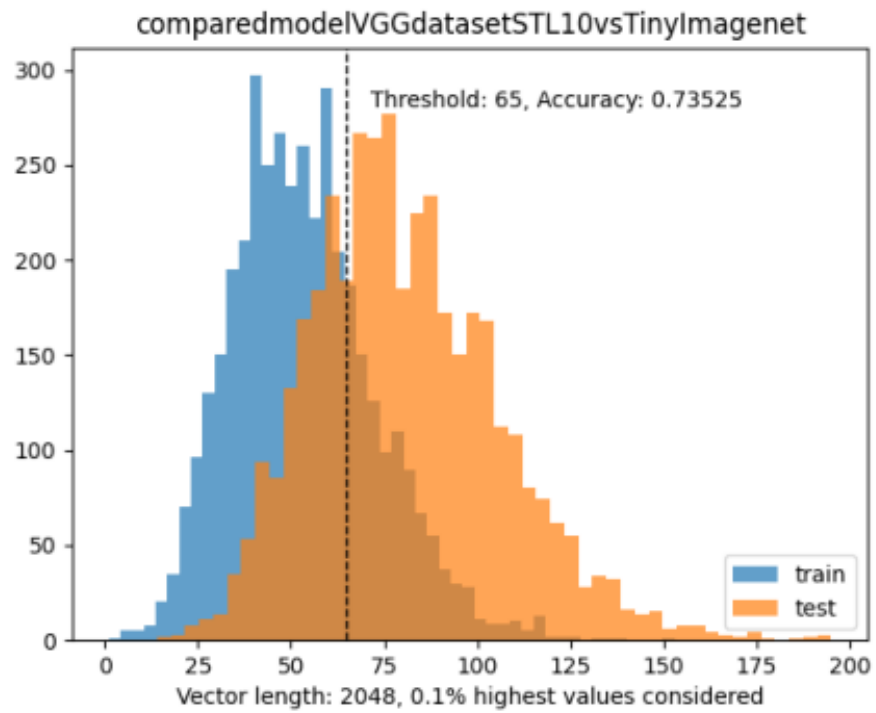


Figure 89: X - dystans; Y - liczba powtorzen.

VGGSTL10vsUniformNoise

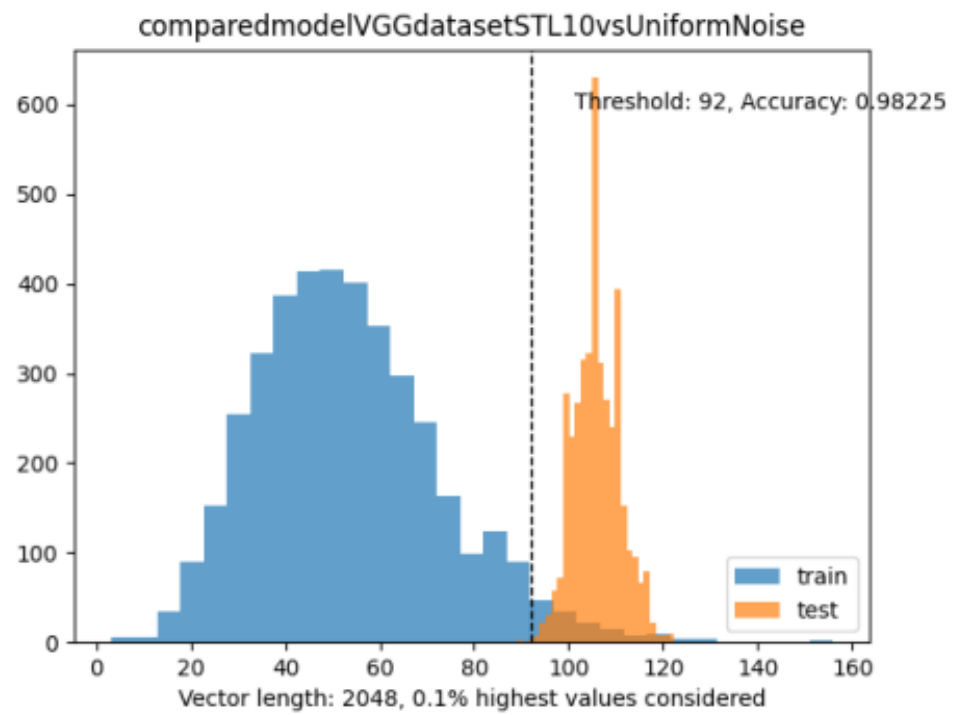


Figure 90: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsCIFAR10

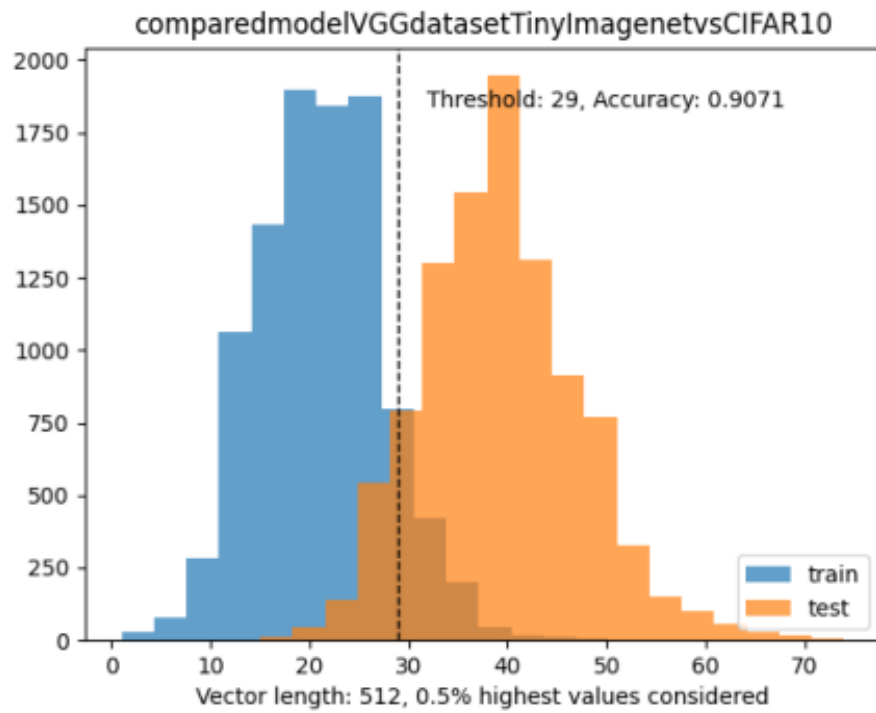


Figure 91: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsCIFAR100

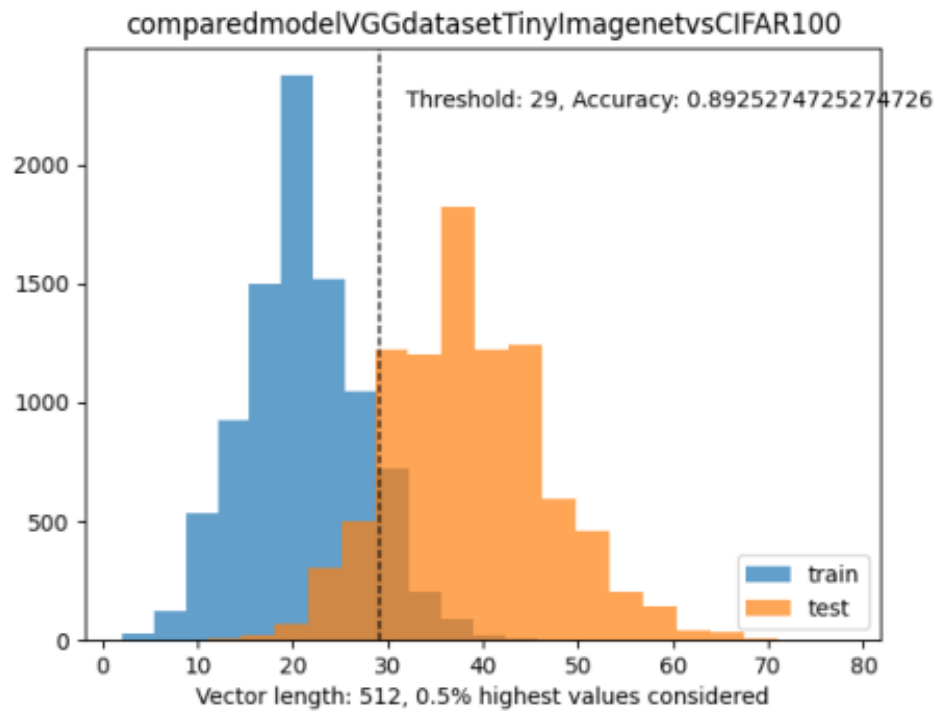


Figure 92: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsFashionMNIST

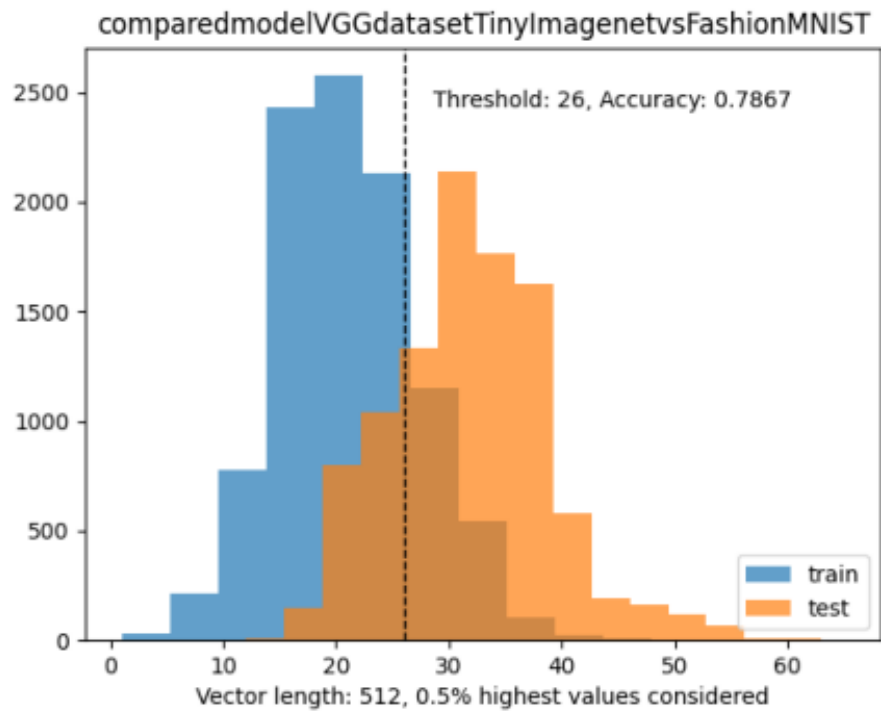


Figure 93: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsMNIST

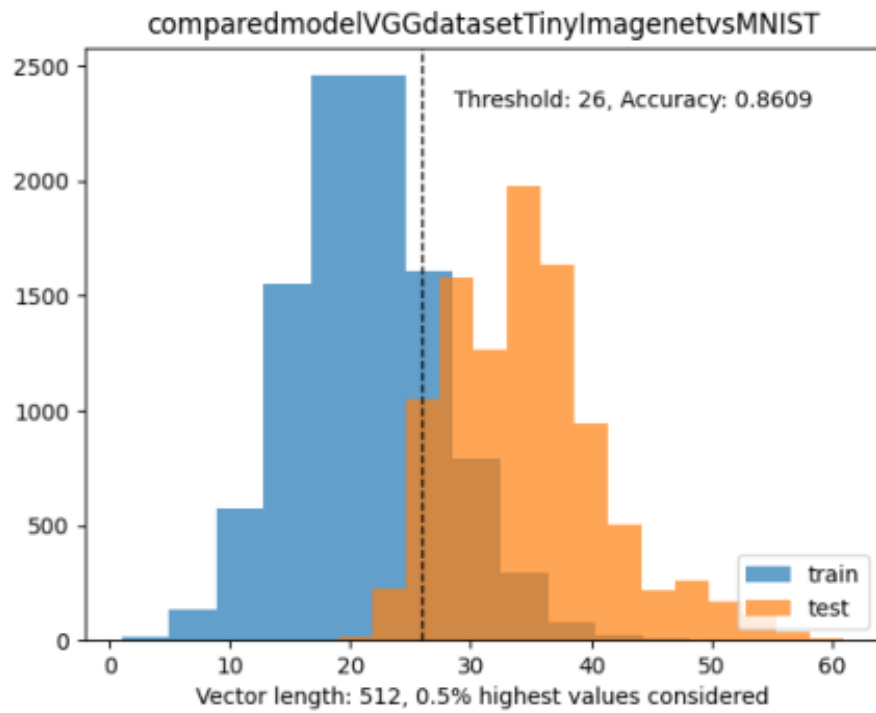


Figure 94: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsNormalNoise

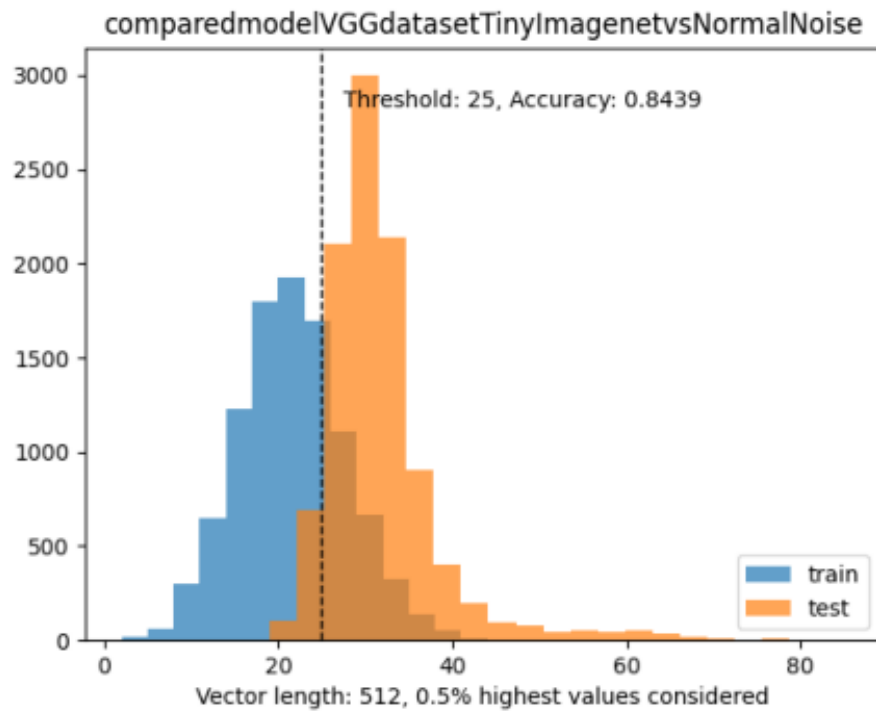


Figure 95: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsNotMNIST

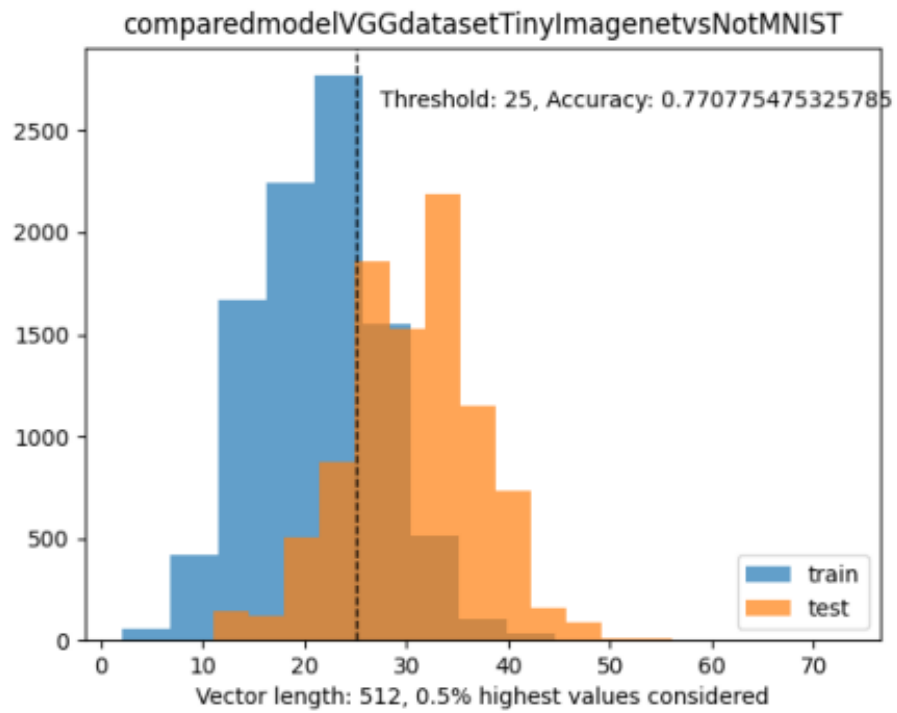


Figure 96: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsSTL10

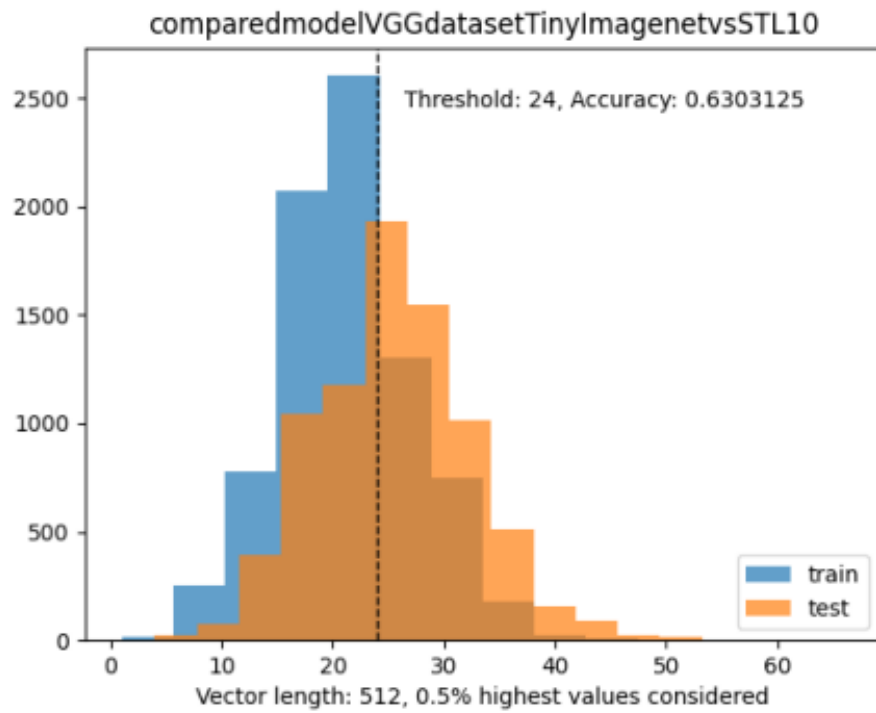


Figure 97: X - dystans; Y - liczba powtorzen.

VGGTinyImagenetvsUniformNoise

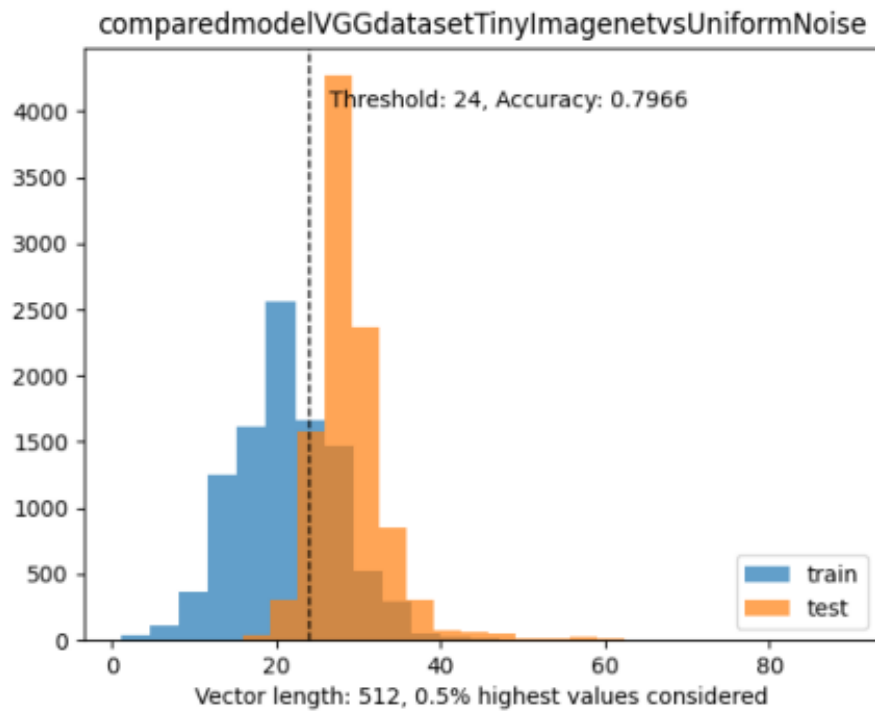


Figure 98: X - dystans; Y - liczba powtorzen.