

## Zadanie 2

Celem zadania było wyuczenie klasyfikatora AdaBoost na wydobytych cechach obrazów, a następnie porównanie wyuczonych modeli w zależności od ilości estymatorów.

Mój proces ekstrakcji cech przebiegał w następującej kolejności: przeskalowywanie do rozmiaru 16x16, całkowanie obrazu, wydobycie cech Haara (bez filtrowania, ponieważ pogarsza to widocznie jakość klasyfikacji co ma znaczenie w późniejszych zadaniach). Ekstrakcja zajęła 31.2s. Zestaw został podzielony na treningowy i walidacyjny w proporcjach 4:1. Na danych treningowych wyuczyłem 3 modele, każdy o innej liczbie estymatorów: 10, 20 i 30. Wyuczenie tych modeli zajęło po kolei 35.9, 71.4 i 106.7 sekund. Widać liniową zależność. Dla zbioru walidacyjnego dostałem miary f1: 0.938, 0.961 i 0.966.

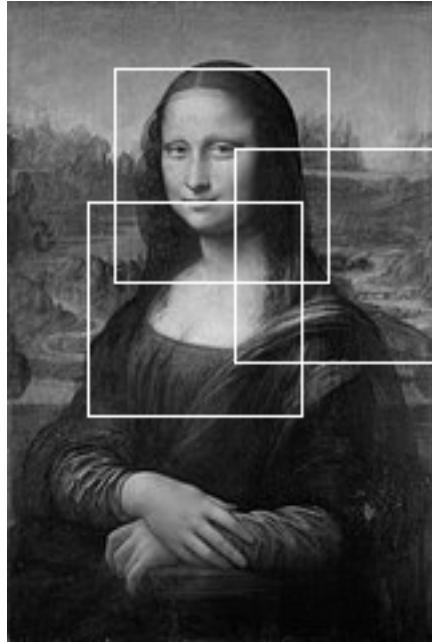
W każdym modelu każde z drzew brało pod uwagę tylko 1 cechę. Sprawdzając te cechy zauważałem, że zbiór cech dla klasyfikatora z 10 estymatorami zbiór cech zawiera się w zbiorach cech klasyfikatorów z 20 i 30 estymatorami. Cechy klasyfikatora z 20 estymatorami też zawierają się w cechach 30-estymatorowego. Oznacza to, że niezależnie od ilości estymatorów, w trakcie uczenia każdy z modeli tak samo ocenił priorytet każdej cechy dla zadania klasyfikacji i według tego dobiera cechy do estymatorów.

## Zadanie 3

Celem tego zadania było wykrycie twarzy na zdjęciach ze zbioru testowego używając wyuczonego modelu z zadania 2.

W moim rozwiązaniu najpierw obraz jest przeskalowywany tak, żeby zadany rozmiar okna z którego wyciągamy cechy został zmieniony w 16x16. Następnie cały obraz jest całkowany. Potem po całym obrazie przesuwane jest okno co ustaloną długość kroku i dla każdej loklizacji obliczane są cechy Haara. Wydobyte cechy są wrzucane do klasyfikatora i wyodrębniane są te lokalizacje okna, dla których klasyfikator zwrócił prawdopodobieństwo klasy 1 przekraczające zadany próg. Użyty jest klasyfikator z 30 estymatorami z zadania 2. Znalezione lokalizacje są nakładane na obraz. Wymagane parametry do uruchomienia detektora to wielkość okna, długość kroku i próg detektora. W szczególności próg detektora musi być strojony ręcznie dla każdego obrazu (co może zająć dużo czasu).

Jak widać jakość metody pozostawia wiele do życzenia, nawet po strojeniu parametrów. Może być to rezultatem zbyt małej ilości cech branych pod uwagę przez klasyfikator, las drzew decyzyjnych nienadający się do tego problemu lub dane treningowe będące niereprezentacyjne dla danych testowych. Póki co nie jestem w stanie określić.



## Zadanie 4

Celem tego zadania było zredukowanie ilości wydobywanych cech do tylko tych, które używane są przez klasyfikator.

Pierwszym krokiem było zmodyfikowanie ekstraktora. Po wyciągnięciu indeksów cech znaczących z klasyfikatora, użyłem `haar_like_feature_coord` żeby zmniejszyć zbiór wydobywanych cech z 32176 do 30. Porównałem działanie poprzedniego ekstraktora zbioru treningowego ze zmodyfikowanym. Czas ekstrakcji spadł z 19.7 do 0.4 sekundy. Taka drastyczna redukcja nie jest niespodzianką, kiedy ilość wyliczanych cech Haara zmalała ponad 1000-krotnie.

Następnie zmodyfikowałem proces wykrywania twarzy na obrazach testowych. Dodałem do niego powyższą modyfikację ekstrakcji cech i mapowałem te cechy do wyzerowanego wektora żeby dało się je włożyć do klasyfikatora. Dla zestawu obrazów testowych zaobserwowałem mniej więcej pięciokrotne przyspieszenie całego procesu i niezmienione wyniki. To też jest oczekiwany rezultat, ponieważ klasyfikator w ogóle nie brał pod uwagę cech poza tymi 30.

## Zadanie 5

Celem zadania było przetestować detekcję twarzy dla klasyfikatorów o różnej liczbie cech i wyłączoną i włączoną optymalizacją cech.



Jak widać na poniższych obrazach zwiększenie ilości cech zmniejsza ilość znalezionych "twarz", ale nie ma widocznej poprawy dokładności. Czasy detekcji były liczone dla całego zestawu 5 zdjęć. Dla pewnego zestawu cech: 84.8s dla modelu z 10 estymatorami, 87.2s dla 20 i 92.3 dla 30. Czasy dla obecnego zestawu cech: 5.1s dla 10, 7.3 dla 20 i 9.8 dla 30. Podobnie jak w poprzednim zadaniu widać drastyczną poprawę czasu działania i takie same rezultaty.

## Zadanie 6

Celem zadania było wykorzystanie modelu bazującego na głębokiej sieci neuronowej do detekcji twarzy na tych samych obrazach i porównanie wyników z wcześniejszymi.

Do tego zadania użyłem modelu MTCNN z biblioteki facenet-pytorch. Dla dokładnego porównania do obliczeń używałem tylko procesora.

Jak widać rezultaty są znacznie lepsze od tych z dotychczasowych modeli. Całość liczyła



się mniej niż sekundę. Może to świadczyć że głębokie sieci neuronowe lepiej nadają się do detekcji twarzy, niż cechy Haara i drzewa decyzyjne.



Rysunek 1: 10 cech

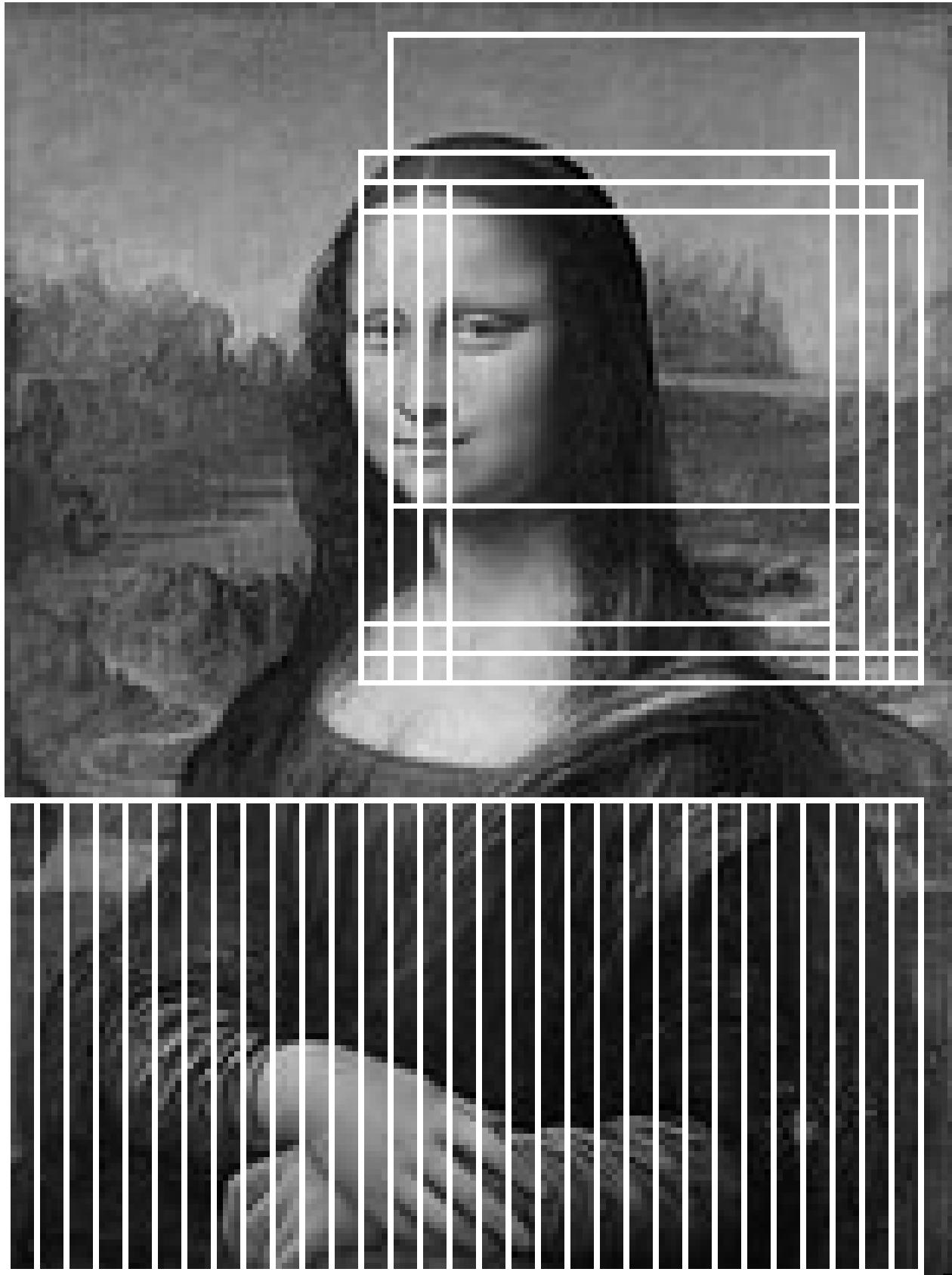


Rysunek 2: 10 cech

Kajetan Bilski 244942

## Sprawozdanie 9

---





Rysunek 4: 10 cech



Rysunek 5: 10 cech



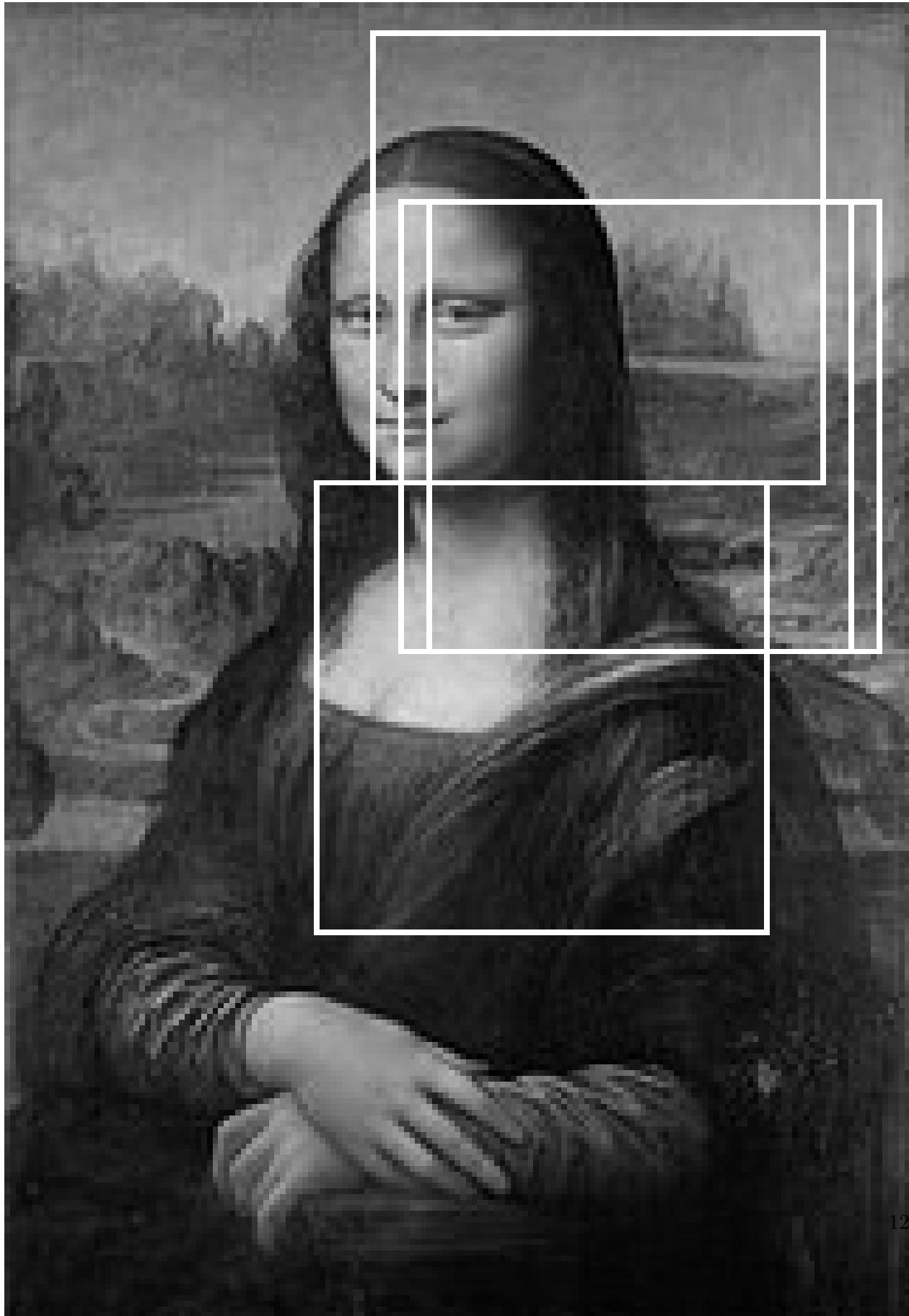
Rysunek 6: 10 cech



Rysunek 7: 20 cech



Rysunek 8: 20 cech





Rysunek 10: 20 cech



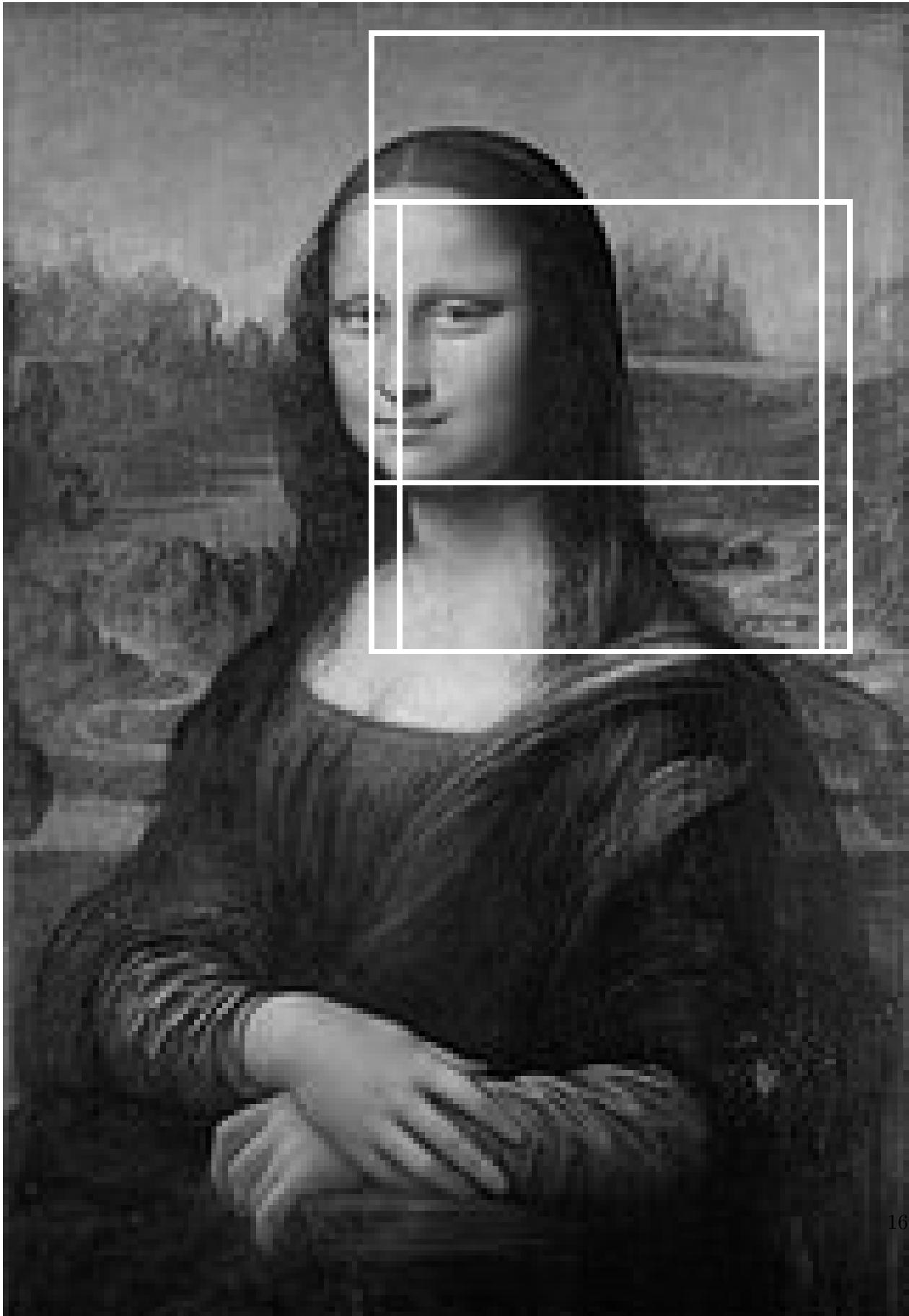
Rysunek 11: 20 cech



Rysunek 12: 30 cech



Rysunek 13: 30 cech

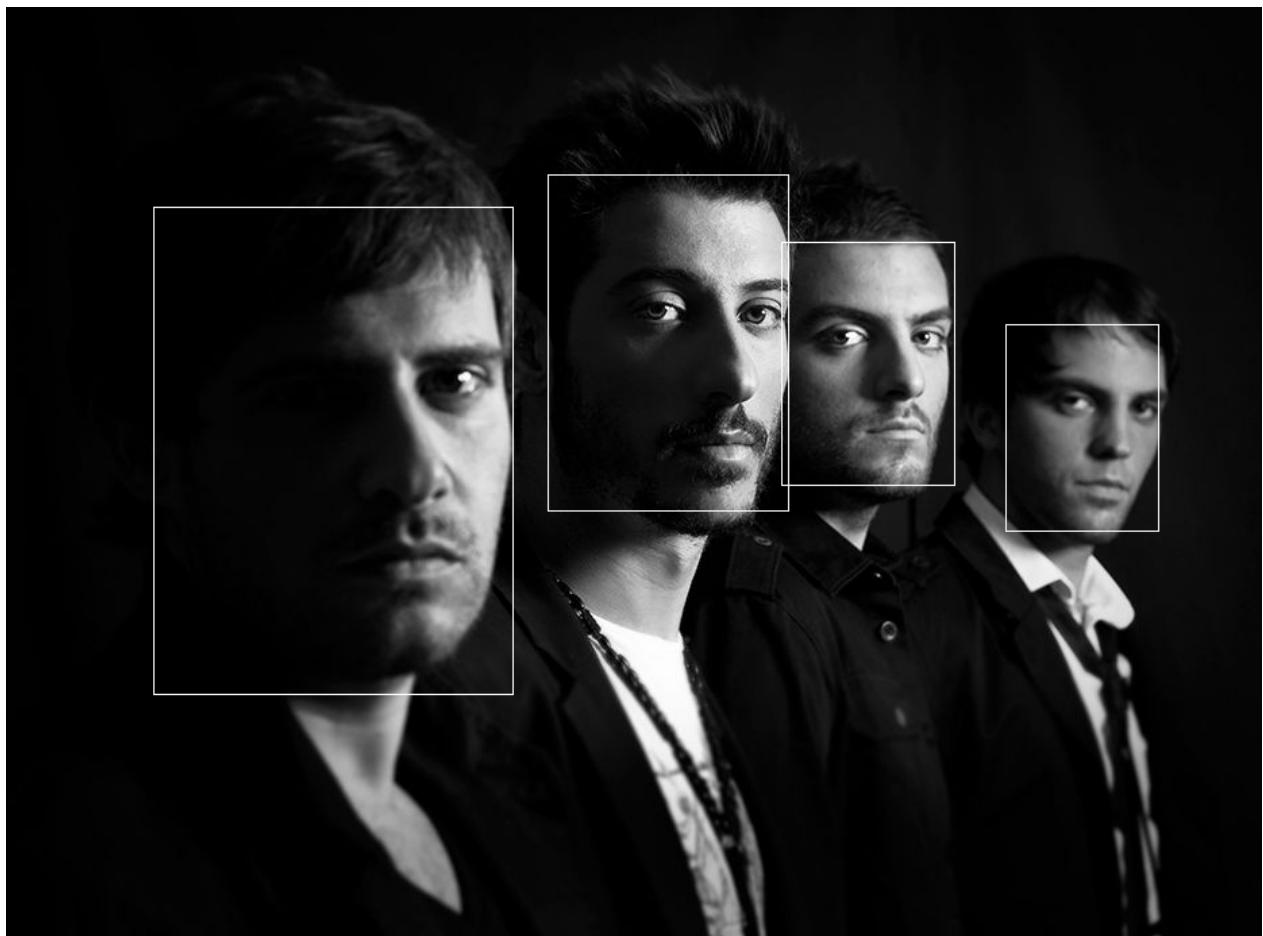




Rysunek 15: 30 cech



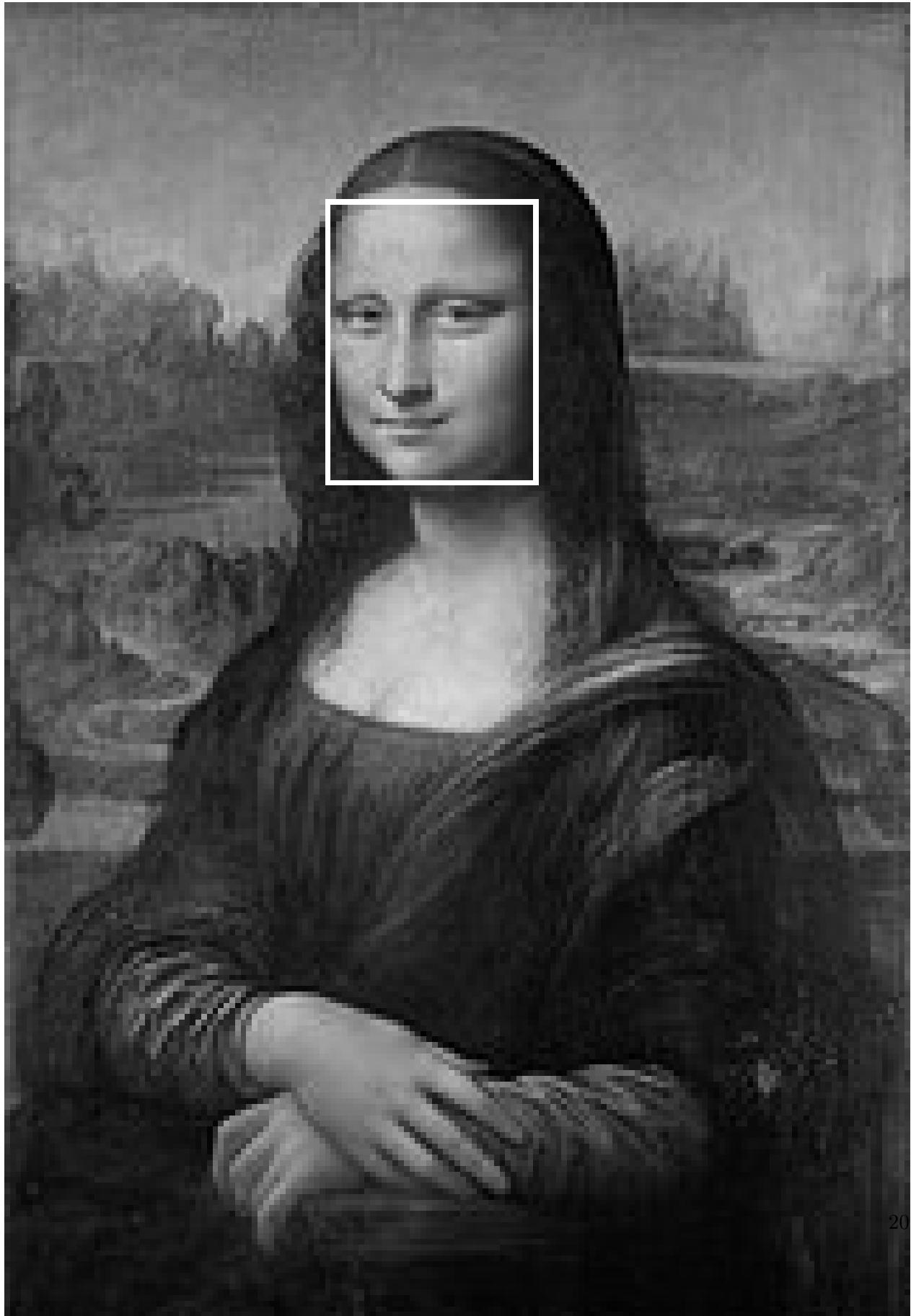
Rysunek 16: 30 cech



Rysunek 17: MTCNN



Rysunek 18: MTCNN





Rysunek 20: MTCNN



Rysunek 21: MTCNN