

Do detekcji cechy binarnej male stworzono model GenderCNN. Na architekturę składały się 4 warstwy konwolucyjne, oddzielone funkcją aktywacji ReLU oraz warstwami próbkowania maksymalnego MaxPool2d, w celu zmniejszenia wymiarowości danych i ograniczenia złożoności obliczeniowej. Na końcu architektury, przed spłaszczeniem do tensora 1D, zastosowano warstwę nn.AdaptiveAvgPool2d(1, 1), która standaryzuje mapy cech do rozmiaru przestrzennego 1×1 . Ostatecznie zastosowano warstwę liniową przekształcającą na wyjście modelu dając informację czy zdjęcie zawierało mężczyznę (1), czy nie (0).

We wszystkich warstwach konwolucyjnych użyto jądra o rozmiarze 3 z padding=1, co pozwala uchwycić drobne szczegóły bez zmiany wymiaru danych. Struktura modelu przedstawia się następująco:

$$3 - 32 - 64 - 128 - 256 - 1$$

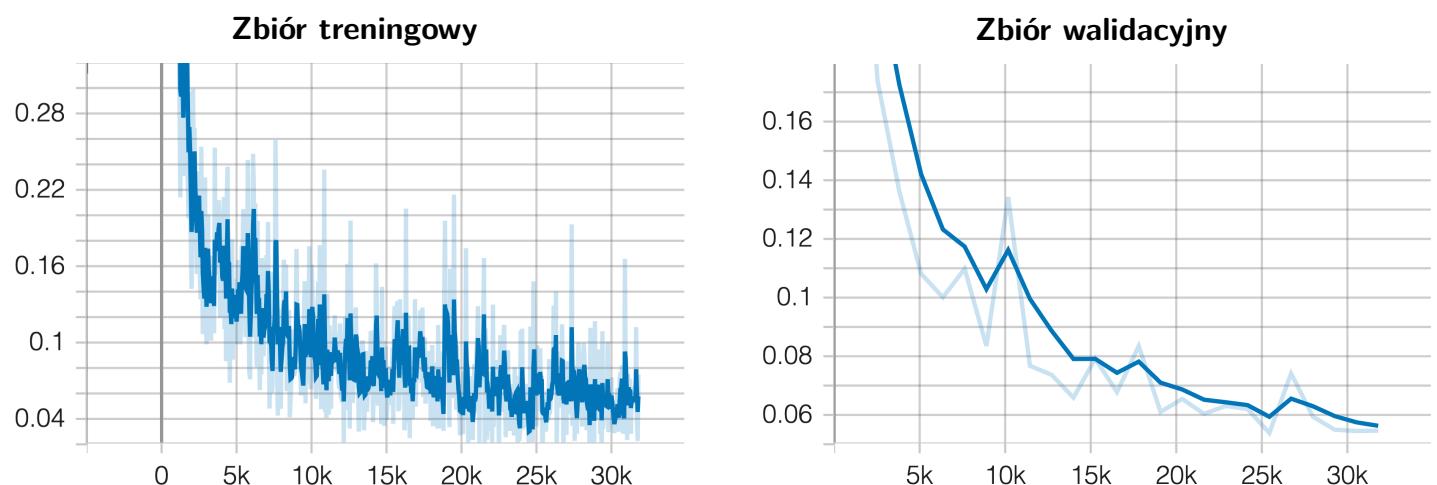
Rozmiar wejściowy równy 3 wynika z faktu, że obrazy w zbiorze CelebA są w kolorze i zawierają trzy kanały kolorów R, G, B. Kolejne rozmiary filtrów dobrano rosnąco w celu umiarkowanego zwiększenia złożoności modelu - wystarczającego do skutecznej ekstrakcji cech, ale bez ryzyka przeuczenia

Proces uczenia oparto o bibliotekę PyTorch Lightning, co zautomatyzowało obsługę pętli treningowej. Jako funkcję celu (*loss function*) zastosowano nn.BCEWithLogitsLoss, łączącą warstwę Sigmoid z *binary cross entropy* dla zapewnienia stabilności numerycznej. Skuteczność modelu monitorowano za pomocą metryki BinaryAccuracy z biblioteki torchmetrics, rejestrując wyniki dla obu zbiorów danych.

Zdjęcia wprowadzane do sieci muszą być przeskalowane na rozmiar 160×160 standaryzując wejścia. Dodatkowo, ze względu na wysoką jakość zdjęć w zbiorze treningowym CelebA, która nie zawsze jest możliwa do osiągnięcia w praktycznych zastosowaniach, zbiór treningowy poddano augmentacji aby lepiej oddać stan realnych sytuacji. Obie te kwestie zostały osiągnięte przez ustawienie potoku augmentacji z wartościami:

- **RandomResizedCrop(160, scale=0.8-1.0)**: przeskalowanie do rozmiaru 160×160 i przybliżanie.
- **RandomHorizontalFlip**: odbicia lustrzane w pionie
- **ColorJitter (jasność=0.2, kontrast=0.2, nasycenie=0.2, odcień=0.1)**: imitacja zmiany oświetlenia
- **Normalizacja**: dane znormalizowanie do zakresu $[-1, 1]$ (średnia 0.5, odchylenie 0.5 dla każdego kanału)

W celu uniknięcia przeuczenia zastosowano procedurę EarlyStopping z parametrem patience=5, monitorującą wartość funkcji celu. Ze względu na niebalansowanie atrybutu male w zbiorze CelebA, do funkcji straty wprowadzono wagę $pos_weight = N_{neg}/N_{pos}$, co wyrównało znaczenie obu klas podczas aktualizacji wag. Poniższy wykres prezentuje przebieg uczenia. Zbieżny spadek funkcji straty na zbiorach treningowym i walidacyjnym oraz stabilizacja metryki accuracy potwierdzają poprawną generalizację modelu i brak przeuczenia przed zakończeniem procesu przez mechanizm EarlyStopping.



Rys. 1: Wykresy przedstawiające spadek wartości funkcji celu w procesie trenowania.

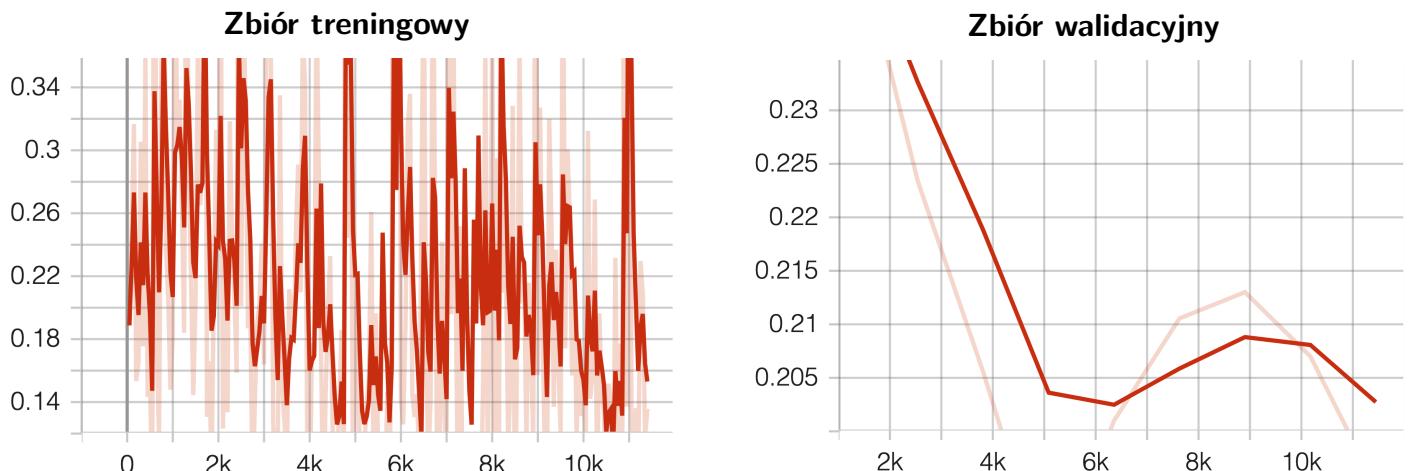
Drugą wybraną cechą do identyfikacji była obecność okularów (atribut `eyeglasses`). Oceną wystąpienia tej cechy zajmował się model `EyeglassesResNet`, w celu stworzenia którego wykorzystano architekturę `ResNet18` wraz z wagami pretrenowanymi na zbiorze `ImageNet` (`IMAGENET1K_V1`). Adaptacja modelu polegała na modyfikacji jego warstwy wyjściowej poprzez zastąpienie oryginalnej warstwy w pełni połączonej (`fc`) nowym elementem liniowym z jednym wyjściem, co umożliwiło dostosowanie sieci do zadania klasyfikacji binarnej.

W procesie uczenia zastosowano mechanizm *fine-tuningu* wszystkich warstw sieci, co pozwoliło na dopasowanie filtrów splotowych do specyfiki zdjęć twarzy przy jednoczesnym zachowaniu wiedzy o ogólnych wzorach wizualnych. Ze względu na wymagania architektury `ResNet`, obrazy wejściowe poddano dodatkowym zabiegom:

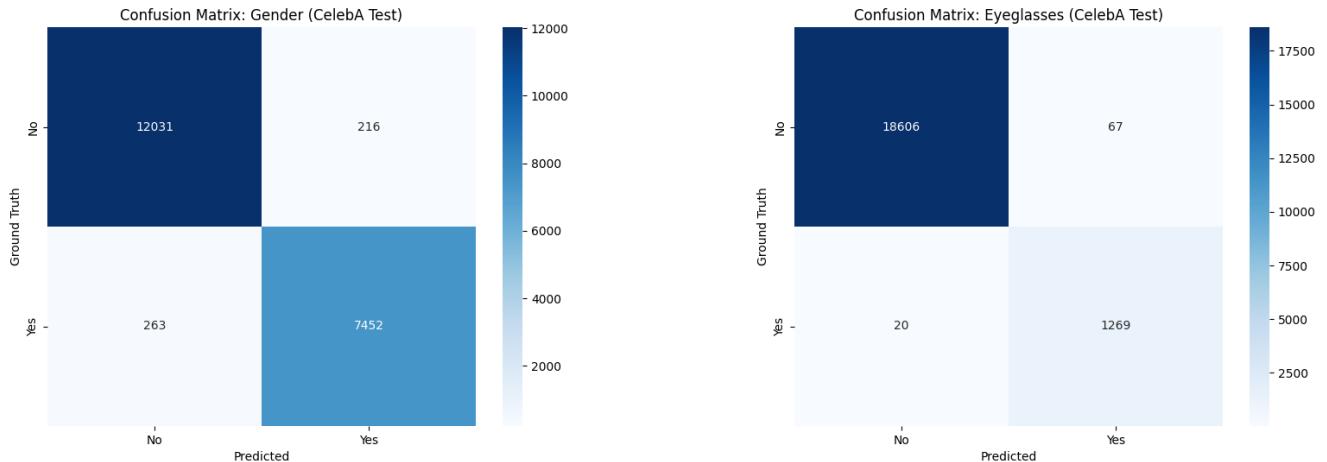
- `CenterCrop(224)`: przeskalowaniu do rozmiaru 224×224 pikseli,
- **Normalizacja**: dane znormalizowane przy użyciu średniej i odchylenia standardowego specyficznego dla wag `ImageNet` ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$).

Pozostałe parametry augmentacji, takie jak losowe odbicia i jitter kolorystyczny, pozostały tożsame z opisanymi przy modelu `GenderCNN`. Trening przeprowadzono z wykorzystaniem optymalizatora `Adam` ($LR = 10^{-3}$) oraz funkcji straty `BCEWithLogitsLoss` przy pomocy modułu `LightningModule`.

Podobnie jak w poprzednim przypadku, zastosowano parametr `pos_weight` w celu niwelowania wpływu niezbalansowania klas w zbiorze `CelebA`. Nad poprawnością procesu czuwał mechanizm *Early Stopping*, przerwający naukę po 5 epokach braku spadku straty walidacyjnej. Tak przygotowany model, po fazie treningu na `CelebA`, został zweryfikowany na dedykowanym zbiorze testowym oraz na wyselekcjonowanej części zbioru `WIDERFace`, dla którego etykiety obecności okularów zostały przygotowane ręcznie.



Rys. 2: Wykresy przedstawiające spadek wartości funkcji celu w procesie trenowania.



Rys. 3: Wykresy przedstawiające macierze pomyłek dla obu stworzonych modeli.

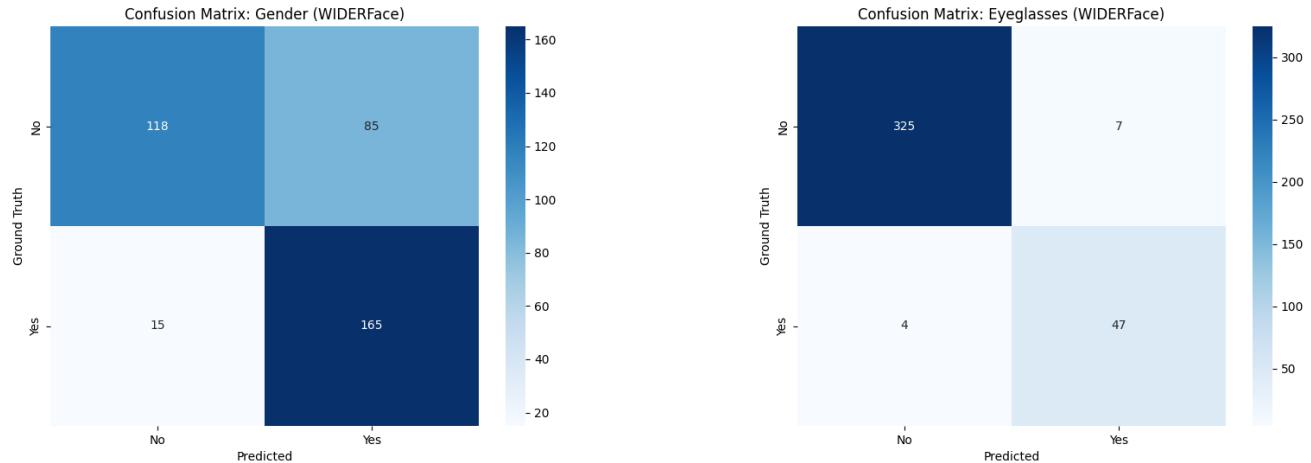
Na podstawie macierzy pomyłek przedstawionych na rys. 3 można zauważyc, że w przypadku atrybutu **Male** model nie wykazuje wyraźnej tendencji do systematycznego błędu w jedną stronę. Z kolei w przypadku atrybutu **Eyeglasses** model częściej błędnie przypisuje obecność okularów, choć różnica ta nie jest duża.

Tabela 1: Przykładowe predykcje dwóch modeli

	TP	TN	FP	FN
GenderCNN				
Prediction	Male	Female	Male	Female
EyeglassesResNet				
Prediction	Glasses	No glasses	Glasses	No glasses

Patrząc na przykładowe wyniki z tabeli 1, można zauważyc, że model prawdopodobnie nauczył się przewidywać atrybut **Male** na podstawie długości włosów, o czym świadczy błędna klasyfikacja mężczyzn jako kobiety. Błędna klasyfikacja kobiety może natomiast wynikać z obecności okularów, ponieważ w zbiorze większość osób w okularach stanowią mężczyźni (972/1289). W przypadku atrybutu **Eyeglasses** przedstawione przykłady są klasyfikowane poprawnie przez model, jednak przypisane w zbiorze etykiety nie potwierdzają poprawności tych przewidywań.

W celu przygotowania testowego zbioru **WIDERFace** zdjęcia zostały najpierw przefiltrowane na podstawie istniejących atrybutów: invalid=0, blur=0, occlusion=0, pose=0 oraz width 50, height 50. Następnie z tak otrzymanego podzbioru ręcznie usunięto obrazy, które jakościowo nie nadawały się do analizy, zastępując je innymi zdjęciami spełniającymi powyższe kryteria. Przygotowano również skrypt pomocniczy, który wyświetlał obrazy i umożliwiał ręczne wprowadzanie wartości atrybutów **Male** oraz **Eyeglasses** po zapoznaniu się ze zdjęciem, a zebrane etykiety zostały zapisane do pliku CSV.



Rys. 4: Wykresy przedstawiające macierze pomyłek dla obu stworzonych modeli.

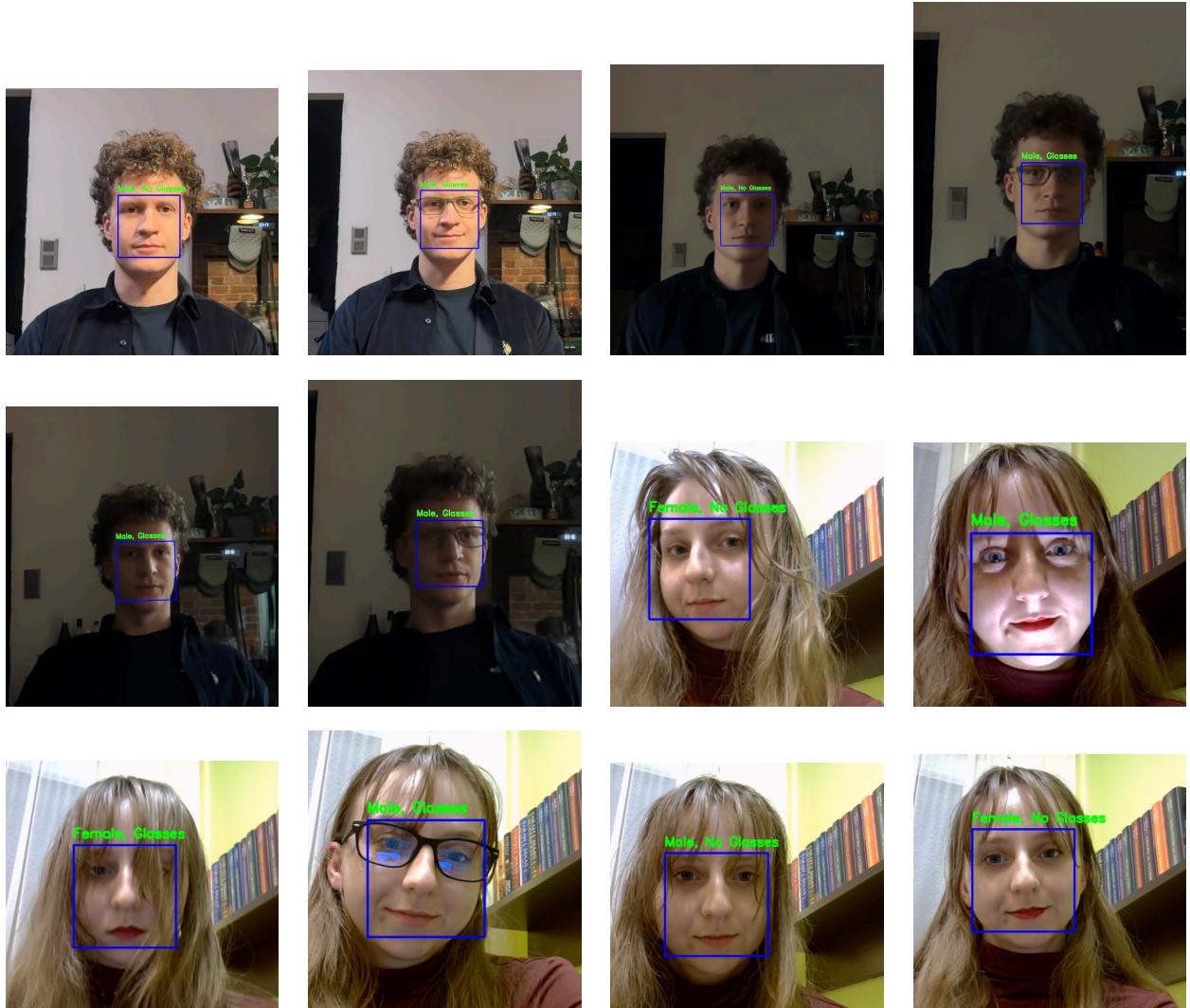
Na podstawie macierzy pomyłek przedstawionych na rys. 4 można zauważyc wyraźną tendencję modelu do błędnej klasyfikacji kobiet jako mężczyzn. Model dla atrybutu **Eyeglasses** cechuje się natomiast wyraźnie mniejszym odsetkiem błędów, bez widocznej dominującej tendencji w którakolwiek stronę.

Tabela 2: Przykładowe predykcje dwóch modeli

	TP	TN	FP	FN
GenderCNN				
Prediction	Male	Female	Male	Female
EyeglassesResNet				
Prediction	Glasses	No glasses	Glasses	No glasses

Patrząc na przykładowe wyniki z tabeli 2, widać, że model prawdopodobnie nauczył się kojarzyć szczupłe twarze kobiet ze zbioru CelebA, co skutkuje błędnymi przewidywaniami. Niepoprawnie zaklasyfikowane przykłady dla atrybutu **Eyeglasses** mogą wynikać ze zbyt słabej jakości zdjęć, na których twarz jest mocno zacieniona, nie widać oprawek okularów.

Zdjęcia na rys. 5 przedstawiają wykorzystanie wytrenowanych modeli w rzeczywistym środowisku. Model kaskadowy wykrywa miejsce, w którym znajduje się twarz, następnie z klatki jest wycinany fragment powiększony o margines wynoszący odpowiednio wysokość i szerokość, tak aby znaczco powiększyć obszar. Wycięty obraz przekazywany jest do modelu odpowiedzialnego za atrybut Male i modelu odpowiedzialnego za atrybut Eyeglasses. Wyniki modeli zapisywane są nad wyświetlana ramką.



Rys. 5: Przykładowe predykcje modelu w rzeczywistym środowisku.

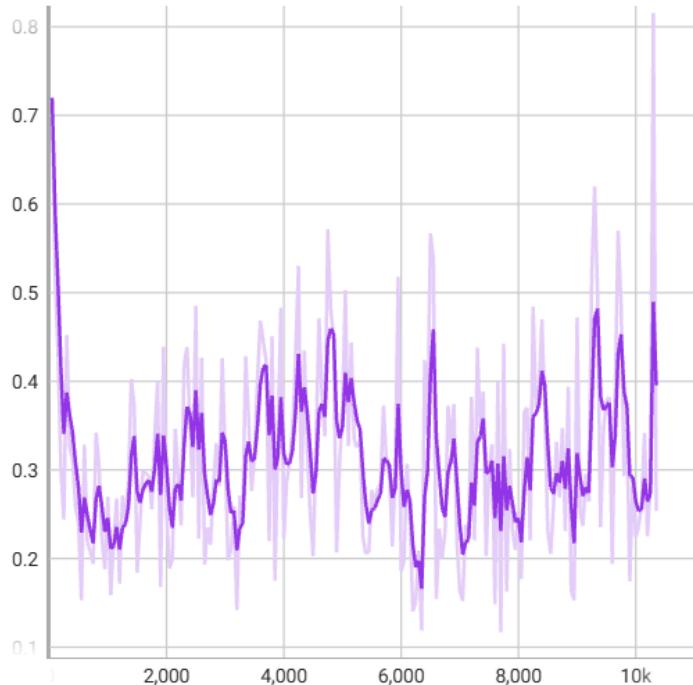
Na przedstawionych zdjęciach widać, że modele stosunkowo dobrze radzą sobie przy różnych warunkach oświetleniowych, a istotne problemy pojawiają się dopiero przy znacznym zaciemnieniu, co skutkuje błędnymi predykcjami atrybutu Eyeglasses. Zauważalne jest również, że model GenderCNN nauczył się wiązać atrybut Male z długością włosów, a zasłonięcie twarzy włosami prowadzi do przewidywania przez model odpowiedzialny za Eyeglasses, że osoba ma okulary. Może to wynikać z faktu, że model interpretuje zaburzenia w obrębie twarzy jako przesłankę do stwierdzenia, że badana osoba nosi okulary. Kwestia na którą warto zwrócić uwagę to mały kontekst detektora, który wpływa na brak uwzględnienia takich cech jak rysy twarzy i w przypadku atrybutu male konieczne było poleganie na makijażu do poprawnej detekcji w większości przypadków.

Do zadania detekcji twarzy stworzono FaceDetectorLightning, do którego wykorzystano dwuetapowy model detekcyjny Faster R-CNN z bazą (*backbone*) MobileNetV3-Large oraz modułem FPN (*Feature Pyramid Network*). Wybór tej architektury podyktowany był koniecznością uzyskania kompromisu między precyją detekcji a wydajnością obliczeniową, co jest kluczowe przy przetwarzaniu obrazu z kamery wideo w czasie rzeczywistym. Adaptacja modelu polegała na modyfikacji modułu predykcyjnego: oryginalny moduł odpowiedzialny za przewidywanie ramek i klas został zastąpiony przez FastRCNNPredictor dostosowany do rozpoznawania dwóch klas (tło oraz twarz). Model zainicjalizowano wagami pretrenowanymi, co pozwoliło wykorzystać cechy wizualne wyuczone na dużych zbiorach danych.

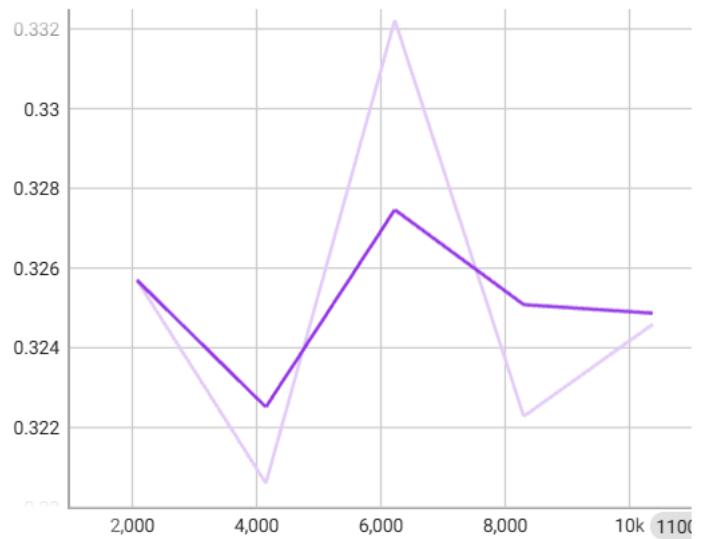
Proces przygotowania danych oparto na zbiorze WIDERFace. Ze względu na bardzo dużą różnorodność próbek (w tym twarze o ekstremalnie niskiej rozdzielcości), wprowadzono procedurę filtrowania w celu poprawy stabilności uczenia. Do treningu dopuszczono jedynie obiekty o wielkości powyżej 30×30 pikseli, które nie były oznaczone jako „invalid” oraz nie charakteryzowały się silnym rozmyciem (parametr blur < 2). Zbiór treningowy został podzielony losowo na część uczącą (80%) oraz walidacyjną (20%). W procesie przygotowania obrazów wykorzystano bibliotekę `torchvision.transforms.v2`, stosując augmentację w postaci losowych odbić poziomych oraz modyfikacji parametrów koloru (*Color Jitter*), co zwiększyło odporność detektora na zmienne warunki oświetleniowe, a także poddano normalizacji jak w poprzednich modelach dla poprawy wyników modelu.

Uczenie zrealizowano przy pomocy optymalizatora AdamW z bardzo niskim współczynnikiem uczenia (10^{-5}), co pozwoliło na precyzyjne dostrojenie wag bez ryzyka ich rozbieżności. Funkcja straty była sumą składowych odpowiedzialnych za klasyfikację obiektów oraz regresję współrzędnych ramek otaczających. Proces monitorowano za pomocą metryki `mAP@50` (*mean Average Precision* przy $IoU = 0.5$). Aby zapobiec przeuczeniu, zastosowano mechanizm *Early Stopping* (`patience=3`) monitorujący stratę walidacyjną oraz *Model Checkpoint* zapisujący parametry sieci dla najwyższej uzyskanej wartości `mAP` na zbiorze walidacyjnym.

Zbiór treningowy



Zbiór walidacyjny



Rys. 6: Wykresy przedstawiające spadek wartości funkcji celu w procesie trenowania.

Stabilizacja wartości funkcji straty (rys. 6) doprowadziła do przerwania procesu uczenia po 4 epokach przez mechanizm *Early Stopping*.

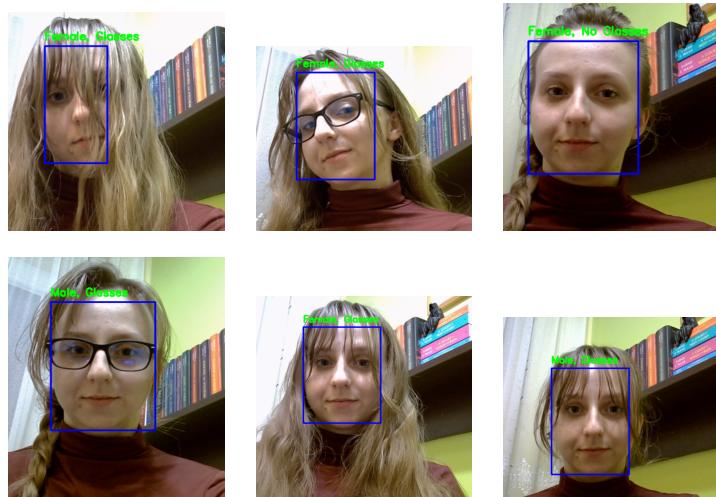
Model osiągnął satysfakcjonującą wartość metryki $mAP_{50} = 0.77$ na zbiorze testowym, co świadczy o wysokiej precyzyji lokalizacji obiektów i skutecznym dopasowaniu ramek otaczających (*bounding boxes*).

Na rysunku 7 pokazano porównanie predykcji (kolor czerwony) z etykietami zbioru WIDERFace (kolor zielony), które potwierdza wysoką skuteczność modelu. Występujące przypadki detekcji twarzy nieoznaczonych w zbiorze (tzw. *False Positives*) wynikają prawdopodobnie z przyjętej strategii akceptacji obrazów o wyższym stopniu rozmycia (*blur*), co pozwoliło modelowi na identyfikację obiektów pominiętych podczas ręcznego etykietowania.



Rys. 7: Przykładowe predykcje modelu FaceDetectorLightning na danych testowych

Wdrożenie detektora CNN wykazało znaczną przewagę nad rozwiązaniem kaskadowym (rys. 8). Model cechuje się wyższą odpornością na trudne oświetlenie, nietypowe kąty nachylenia głowy oraz częściowe zasłonięcie twarzy, co dla modelu kaskadowego stanowiło barierę krytyczną. Stabilniejsza lokalizacja obiektów bezpośrednio wpłynęła na pewność predykcji modeli GenderCNN i EyeglassesResNet. Na rysunku 8 widać, że dzięki precyzyjnemu kadrowaniu etykiety atrybutów są stabilne w czasie rzeczywistym, co potwierdza wysoką skuteczność całego potoku przetwarzania w warunkach naturalnych.



Rys. 8: Przykładowe predykcje modelu FaceDetectorLightning w rzeczywistym środowisku.