



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
NAZWA WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII
BIOMEDYCZNEJ

Informatyka w sterowaniu i zarządzaniu
Głębokie uczenie i inteligencja obliczeniowa
Grupa 2 (śr 13:30 – 15:45)

Rozpoznawanie owoców i warzyw z wykorzystaniem splotowych sieci neuronowych oraz analiza mapy aktywacji

L.p.	Członek	Numer albumu	Adres e-mail
1	Patryk Chorąży	402569	pchorazy@student.agh.edu.pl
2	Rafał Kośla	400332	rkosla@student.agh.edu.pl
3	Artur Mzyk	400658	arturmzyk@student.agh.edu.pl
4	Joanna Nużka	400561	joannanuzka@student.agh.edu.pl
5	Adrian Poniatowski	401346	adrianponiat@student.agh.edu.pl
6	Wojciech Poniewierka	402224	wponiewierka@student.agh.edu.pl

Spis treści

1.	Wstęp	3
1.1.	Cel projektu	3
1.2.	Założenia projektu.....	3
2.	Badany problem.....	3
3.	Propozycja rozwiązania.....	3
4.	Aplikacja.....	4
5.	Eksperymenty	5
5.1.	ResNet50	5
5.2.	VGG16	15
5.3.	Porównanie czasu uczenia sieci	25
5.4.	Poprzedni dataset.....	26
6.	Wnioski.....	26
7.	Literatura	27
8.	Podział pracy	27

1. Wstęp

1.1. Cel projektu

Celem projektu jest klasyfikacja owoców i warzyw z wykorzystaniem splotowych sieci neuronowych (ang. *Convolutional Neural Networks*) oraz analiza mapy aktywności (ang. *Class Activation Mapping – CAM*) w celu odtworzenia toku rozumowania sieci neuronowej.

1.2. Założenia projektu

Zbiór danych stanowi 12000 obrazów z ogólnodostępnej bazy ze zdjęciami świeżych lub zepsutych owoców i warzyw [1]. Są one podzielone na 20 różnych rodzajów. Na jednym zdjęciu może występować jedna lub kilka sztuk danego owocu czy warzywa, zaś tło może być niejednolite.

2. Badany problem

Badanym problemem jest klasyfikacja owoców i warzyw – przypisanie obrazowi odpowiedniej etykiety na podstawie wyjścia sztucznej sieci neuronowej. Obrazy przekazywane są jako macierze trójwymiarowe, gdzie pierwsze dwa wymiary opisują rozmiar obrazu (w naszym przypadku 150 x 150), a trzeci wymiar – wartości piksela w przestrzeni RGB. Etykiety są ciągiem znaków – angielską nazwą danego owocu bądź warzywa.

3. Propozycja rozwiązania

Zastosowane zostaną splotowe sieci neuronowe, którą są świetnym narzędziem do ekstrahowania cech z obrazów, gdyż skutecznie wydobywają cechy w lokalnych obszarach i redukują rozmiar obrazu w kolejnych warstwach sieci. Rozpoznawanie owoców nastąpi głównie przez wzgląd na takie parametry jak kształt czy kolor.

Wykorzystane zostaną gotowe architektury sieci splotowych, takie jak VGG (ang. *Visual Geometry Group*) oraz ResNet (*Residual Neural Network*). W takich architekturach zostaną zmienione ostatnie warstwy, aby lepiej dostosować się do rozważanego zagadnienia, oraz przeprowadzone zostanie douczanie sieci na pobranym zbiorze danych – wykorzystanie techniki *transfer learning*.

Zostanie przeprowadzona również analiza rozumowania sieci poprzez zastosowanie map aktywacji CAM, które wskażą, na którym obszarze obrazu najbardziej skupiała się splotowa sieć neuronowa.

4. Aplikacja

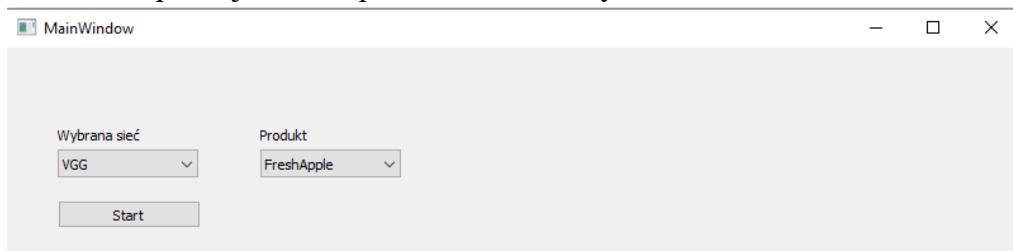
Projekt napisaliśmy w języku python z wykorzystaniem modeli sieci dostępnych w bibliotece tensorflow.keras.

Zbudowany został interfejs graficzny, który umożliwia wgranie zdjęcia, na którym znajdują się owoce lub warzywa. Wykorzystuje on już nauczone modele sieci neuronowych. W wyniku uruchomienia aplikacji otrzymujemy:

- etykiety owoców i warzywa rozpoznanych na obrazie wraz z ich prawdopodobieństwem,
- mapę aktywacji.

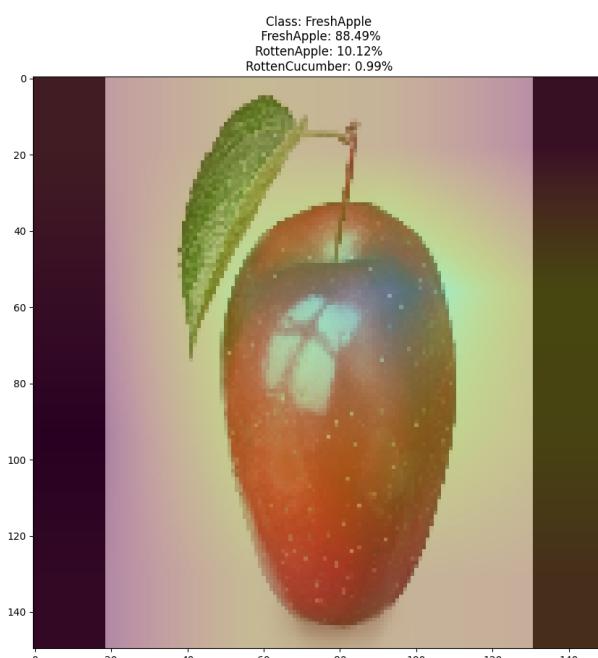
Przed uruchomieniem aplikacji należy ustawić ścieżkę do wytrenowanego modelu w pliku *config.json*. Sieć można wytrenować w jednym z notebooków: *vgg.ipynb* lub *resnet.ipynb* albo użyć gotowego pliku – *model_vgg16.h5*. Następnie należy uruchomić plik *main.py*.

Główne okno aplikacji zostało przedstawione na rysunku 1.



Rysunek 1. Główne okno aplikacji

W oknie tym istnieje możliwość wyboru sieci pomiędzy VGG oraz ResNet po wcześniejszym podaniu ścieżek do nich w pliku *config.json*. Należy także wybrać etykietę produkty, który planujemy rozpoznawać. Wykorzystywane jest to potem do wyświetlenia prawidłowego podpisu w oknie z rezultatami (rys. 2).



Rysunek 2. Rezultat działania aplikacji

Po kliknięciu przycisku start w głównym oknie ukazuje się okno, w którym trzeba wybrać plik - zdjęcie z obiektem do rozpoznania. Następnie następuje predykcja, której wyniki są wypisane nad zdjęciem, natomiast zdjęcie oprócz oryginalnego obrazu zawiera także nałożoną mapę aktywacji.

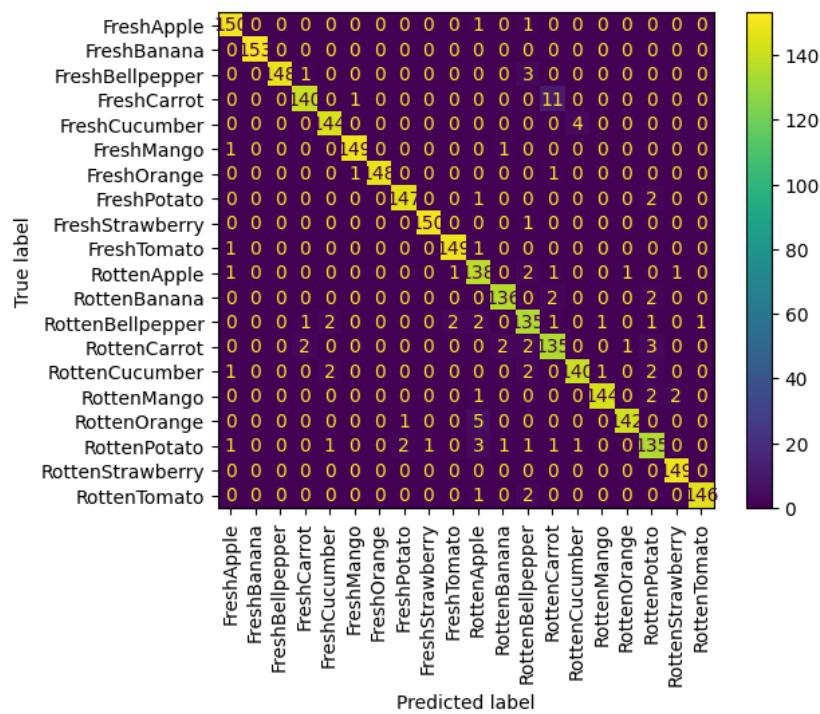
5. Eksperymenty

Dla każdej z sieci badaliśmy wpływ różnych modyfikacji – odmrożenia podczas nauki 1, 2, lub trzech ostatnich warstw, a także dodania naszych warstw. Każda z sieci była trenowana przez 50 epok.

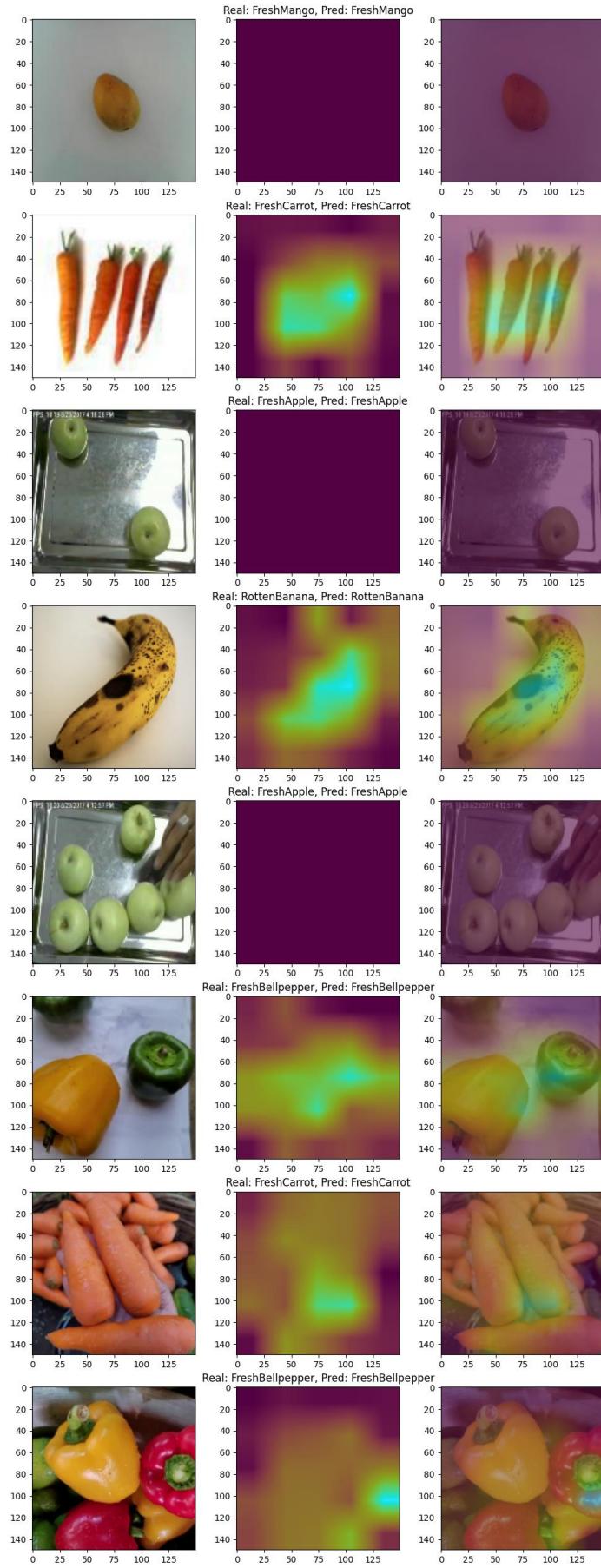
5.1. ResNet50

Podstawowa sieć

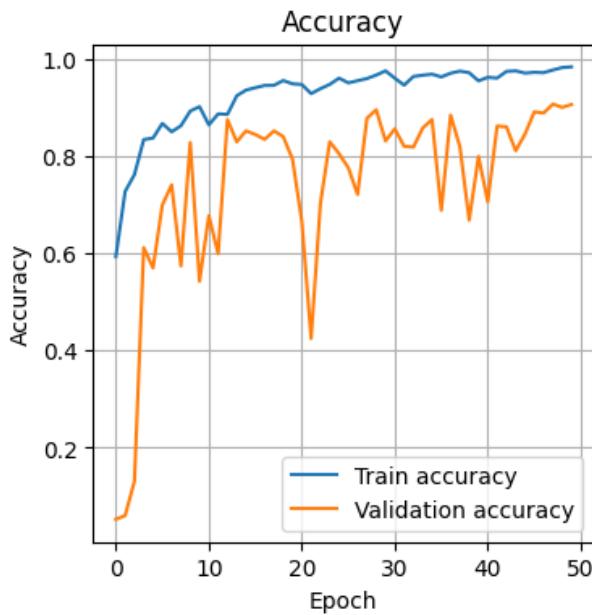
Dla podstawowej sieci ResNet50 uzyskaliśmy dokładność dla zbioru testowego na poziomie 96,7%.



Rysunek 3. Macierz pomylek



Rysunek 4. Analiza GradCAM

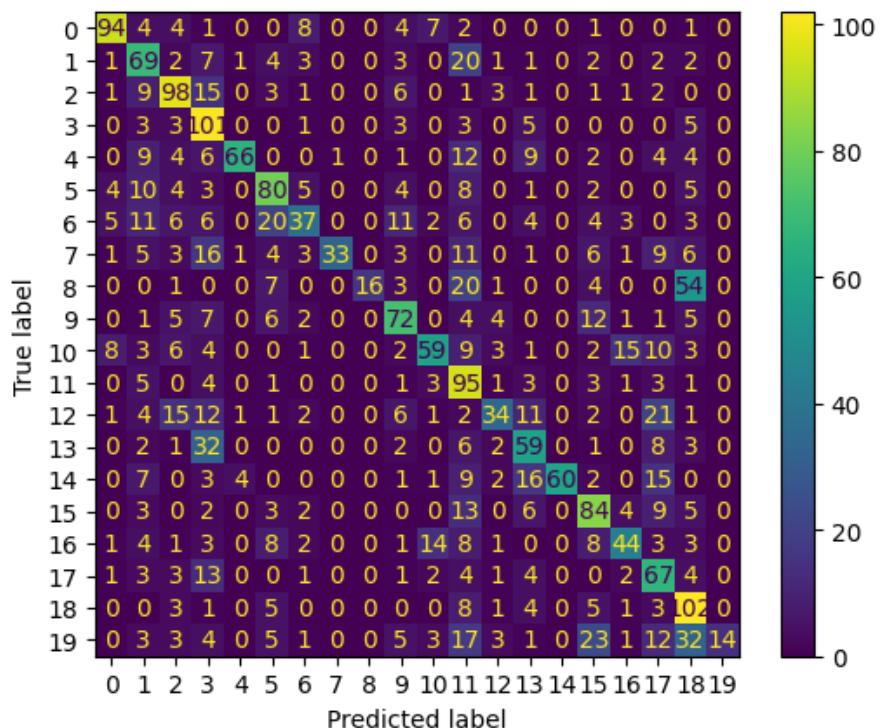


Rysunek 5. Dokładność w kolejnych epokach

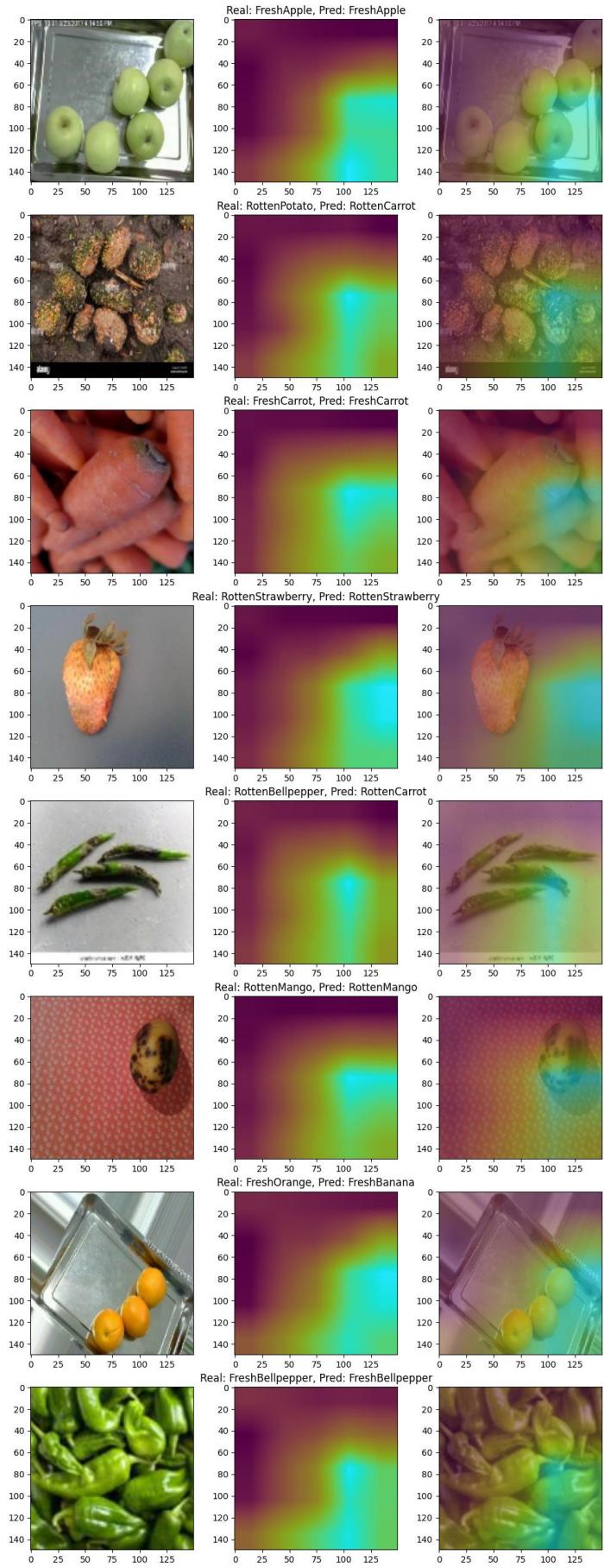
Sieć ResNet50 w podstawowej wersji poradziła sobie całkiem dobrze z zagadnieniem rozpoznawania owoców i warzyw osiągając dokładność na poziomie 96,7%. Analiza CAM pokazuje, że w przeważającej części rozpoznawanych obrazów sieć brała pod uwagę obszary, w których znajdował się obiekt. W tym przypadku największy problem stanowiło rozróżnienie obiektów FreshCarrot oraz RottenCarrot.

Odmrożona jedna warstwa

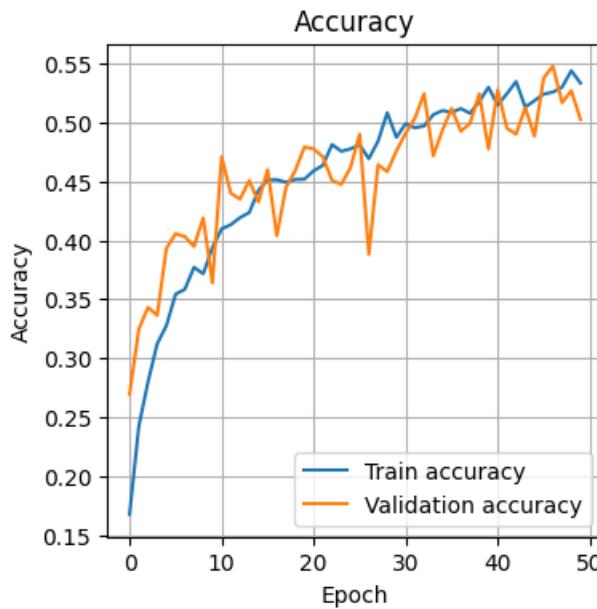
Dla sieci ResNet50 i jednej odmrożonej warstwy uzyskaliśmy dokładność dla zbioru testowego na poziomie 54%.



Rysunek 6. Macierz pomylek



Rysunek 7. Analiza GradCAM

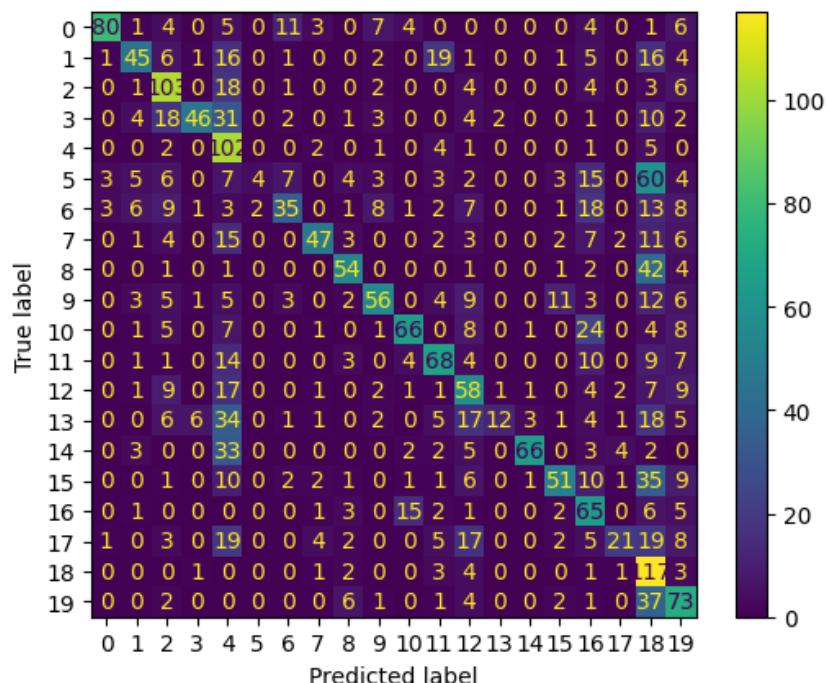


Rysunek 8. Dokładność w kolejnych epokach

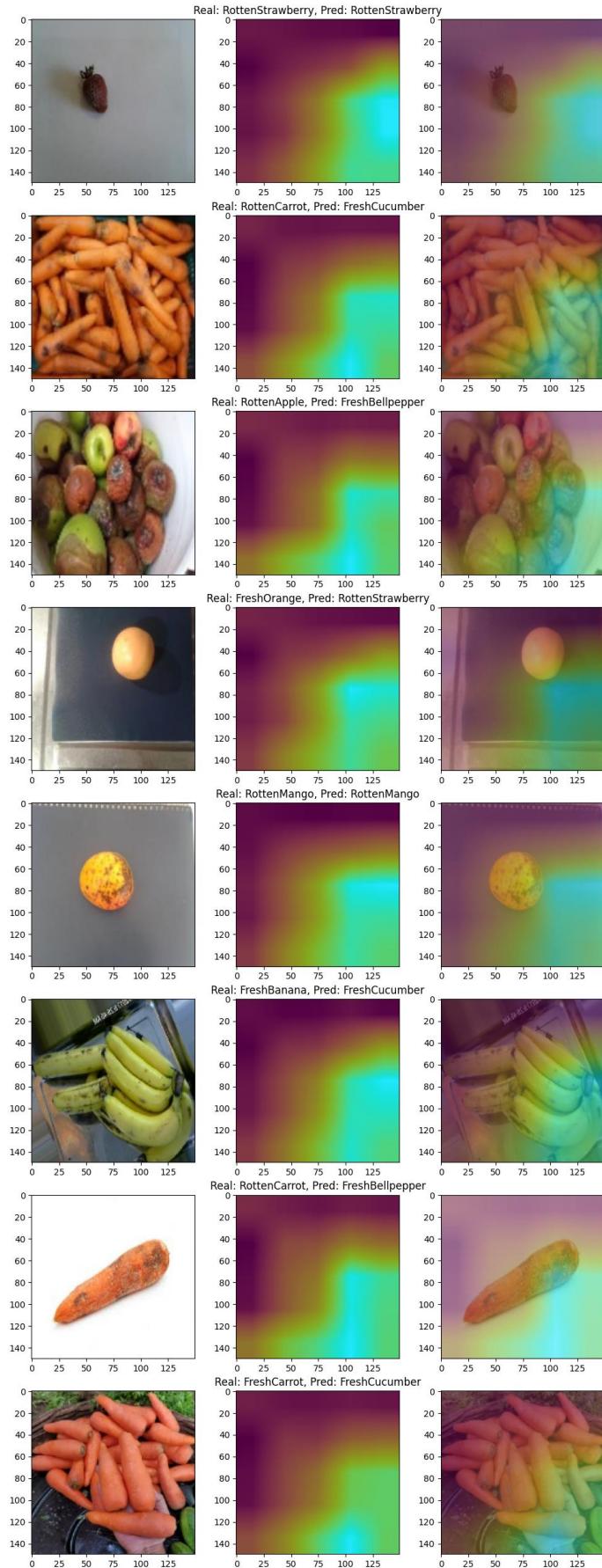
Sieć uzyskała bardzo słabe rezultaty mimo dość długiego czasu trenowania. Na podstawie macierzy pomyłek można zauważać, że dla niektórych etykiet sieć podawała błędne wartości etykiet zdecydowanie częściej niż poprawne. Na podstawie analizy CAM można zauważać, że sieć brała pod uwagę różne obszary obrazu, jednak nie zawsze znajdował się tam obiekt. Czasem sieć znajdowała odpowiedni obszar obrazu, ale obiekt zostawał błędnie rozpoznany.

Odmrożone dwie warstwy

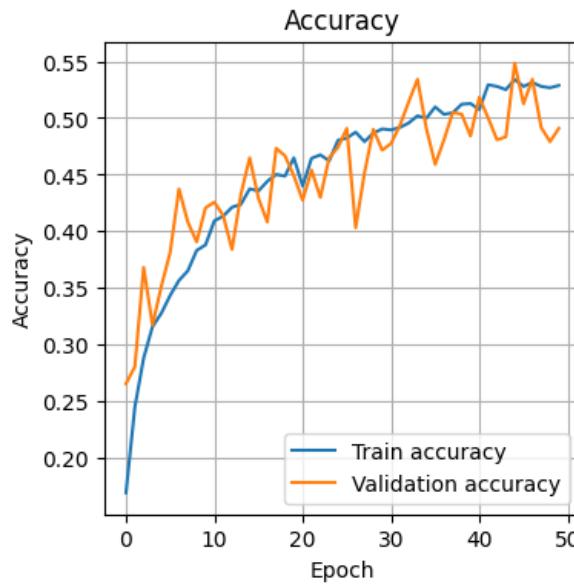
Dla sieci ResNet50 i jednej odmrożonej warstwy uzyskaliśmy dokładność dla zbioru testowego na poziomie 49%.



Rysunek 9. Macierz pomyłek



Rysunek 10. Analiza GradCAM

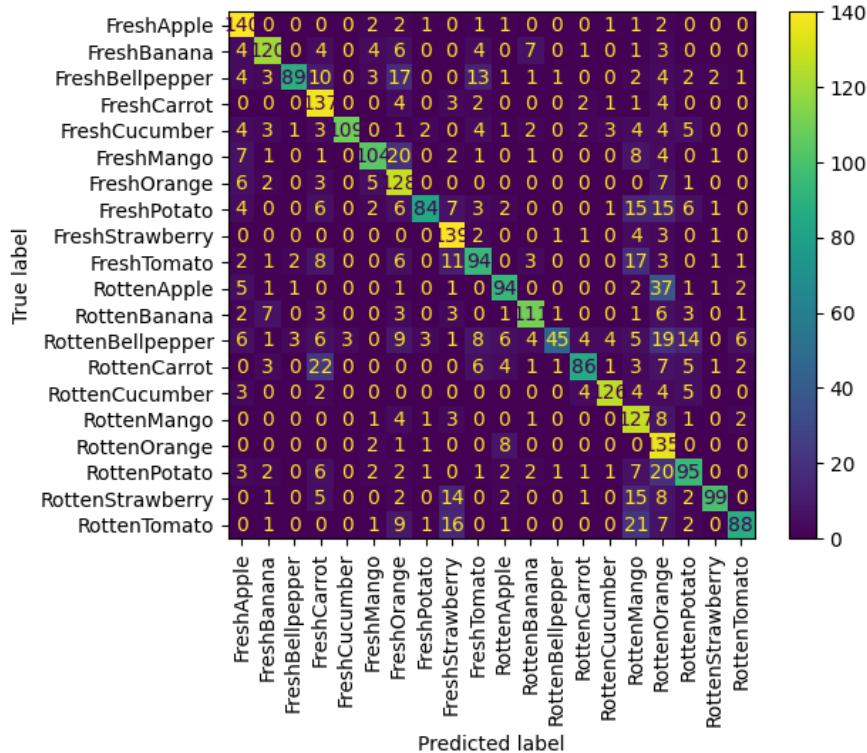


Rysunek 11. Dokładność w kolejnych epokach

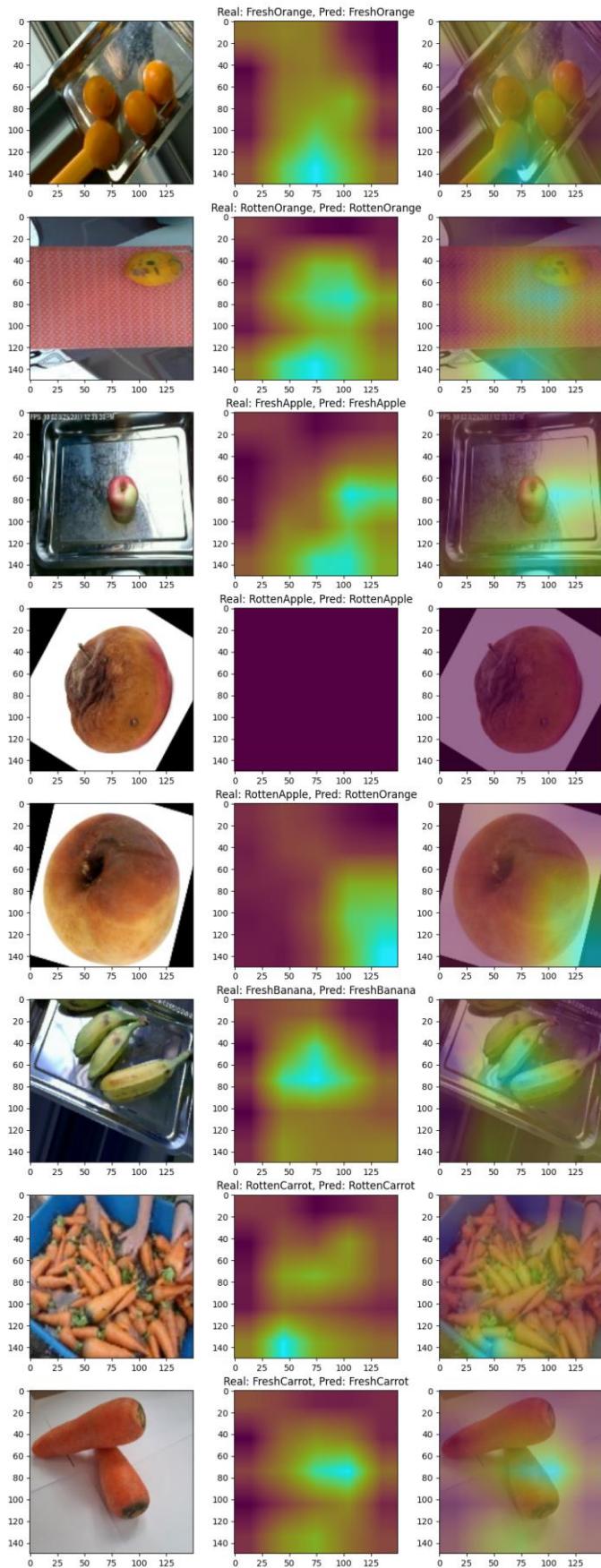
Podobnie jak w przypadku jednej odmrożonej warstwy sieć uzyskała bardzo słabe rezultaty i dokładność dla zbioru testowego na poziomie poniżej 50%. Wskazuje na to również macierz pomyłek oraz analiza CAM. Pokazuje ona, że sieć nie do końca potrafi znaleźć obszar, w którym znajduje się obiekt i często podejmuje decyzję na podstawie tła.

Odmrożone trzy warstwy

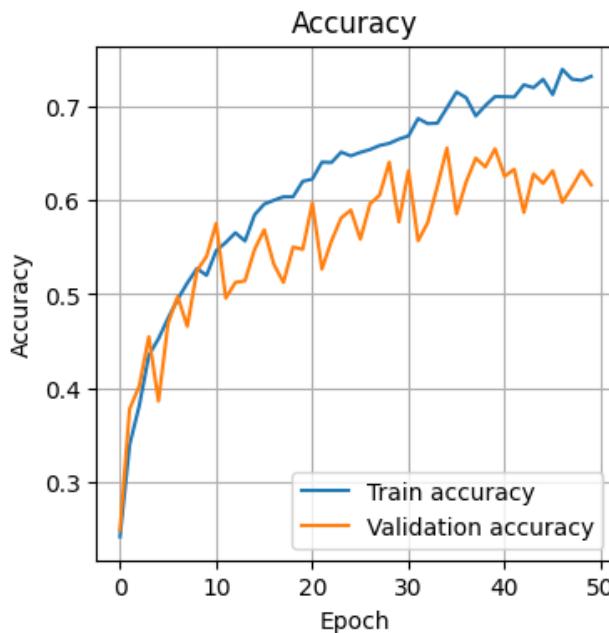
Dla sieci ResNet50 i trzech odmrożonych warstw udało się uzyskać dokładność dla zbioru testowego na poziomie 72,2%.



Rysunek 12. Macierz pomyłek



Rysunek 13. Analiza GradCAM

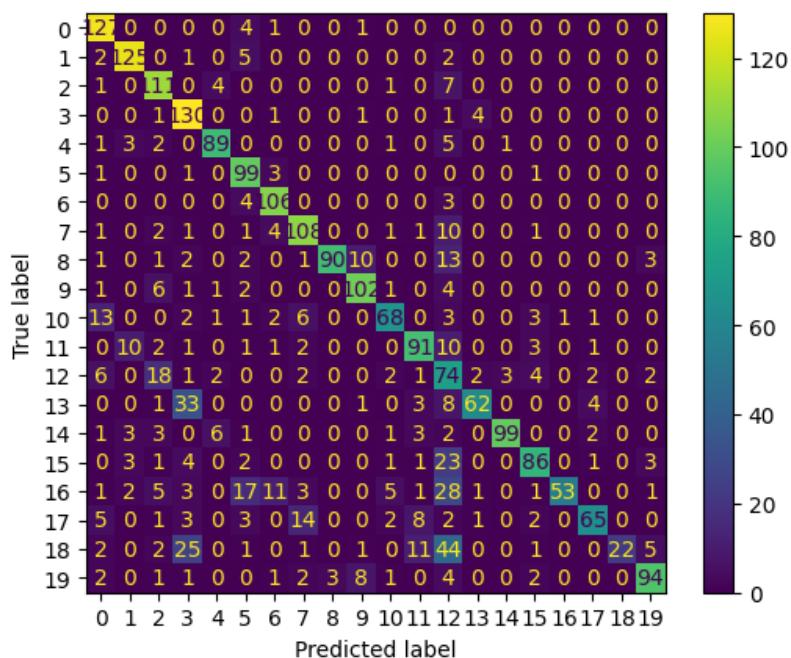


Rysunek 14. Dokładność w kolejnych epokach

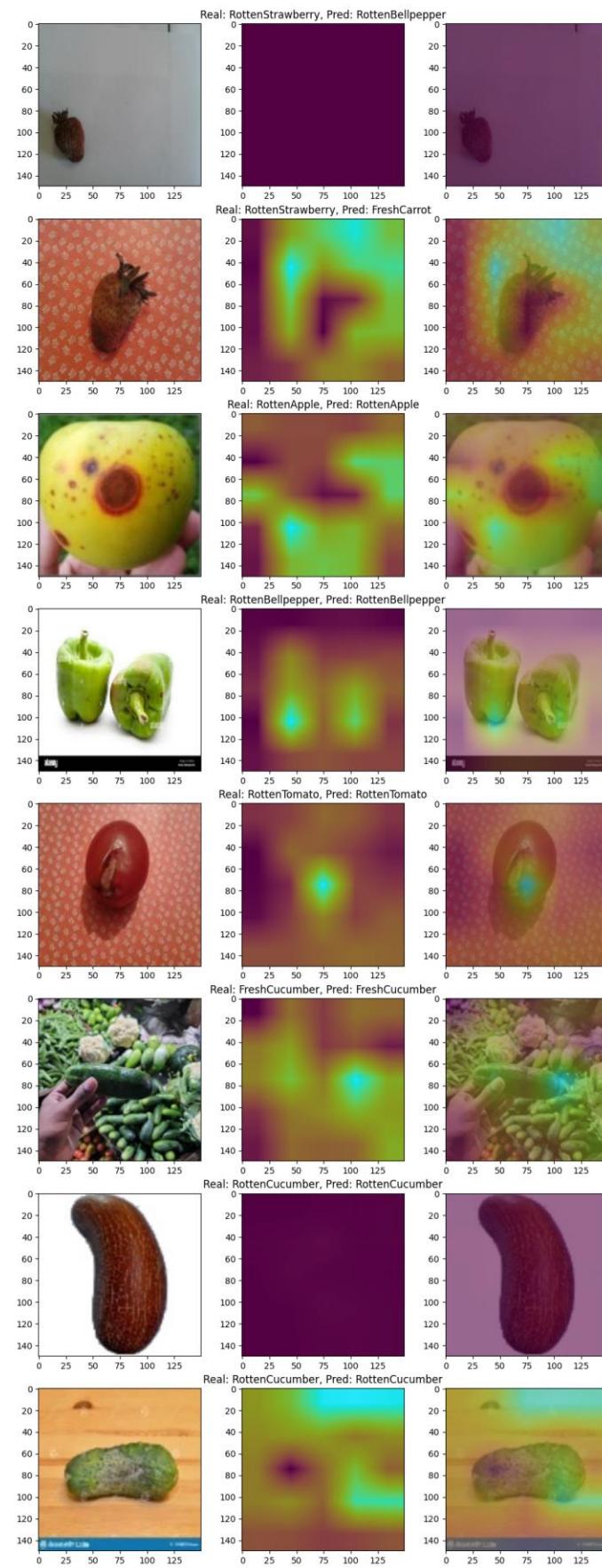
Na rys. 12 można zaobserwować, że RottenApple zostało sklasyfikowane jako RottenOrange, co jest najczęstszym błędem - występuje aż 37 razy na macierzy pomyłek. W przypadku odmrożonych trzech warstw otrzymaliśmy lepszą dokładność niż w przypadku jednej lub dwóch odmrożonych. Są to jednak znacznie niższe wyniki niż w przypadku sieci ResNet50 w podstawowej wersji.

Dodanie dodatkowych warstw

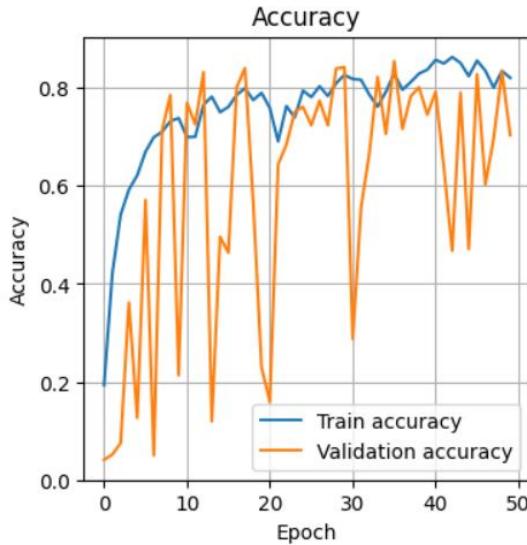
Dla sieci ResNet50 i dodania własnych warstw: relu + dropout + relu + dropout otrzymaliśmy dokładność dla zbioru testowego na poziomie ok. 75%.



Rysunek 15. Macierz pomyłek



Rysunek 16. Analiza GradCAM



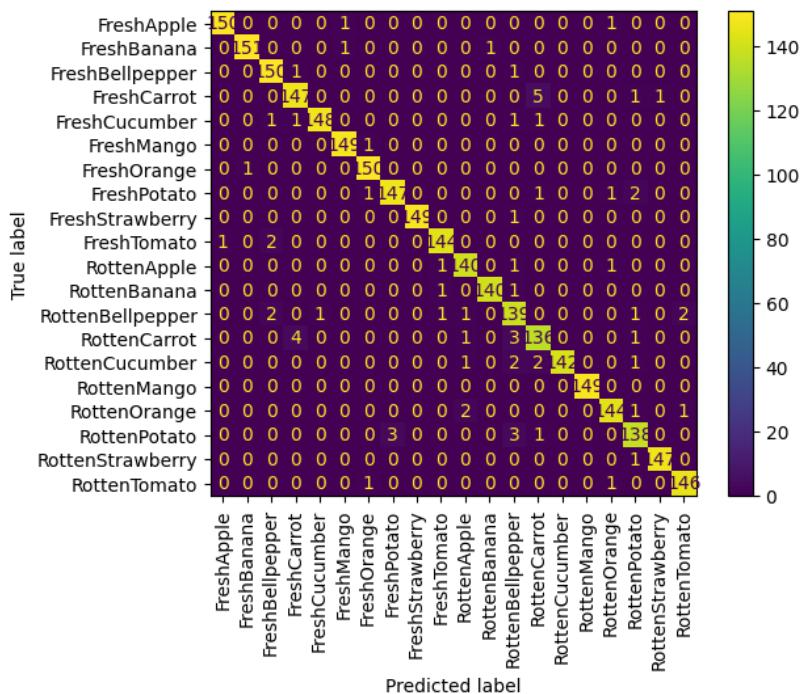
Rysunek 17. Dokładność w kolejnych epokach

Po dodaniu własnych warstw otrzymaliśmy lepszą dokładność niż dla odmrażania warstw, ale gorszą niż dla bazowej sieci ResNet50. Wykres dokładności w kolejnych epokach pokazuje bardzo duże jej wahania dla zbioru walidacyjnego: od ok. 10 do 80%. Oscylacje zmniejszają się jednak w kolejnych epokach, sieć ta może więc potrzebować ich więcej, aby się dobrze nauczyć. Na podstawie analizy GradCAM widzimy, że sieć zazwyczaj dobrze wskazywała obiekty na obrazach, chociaż czasem miała też problem z ich znalezieniem.

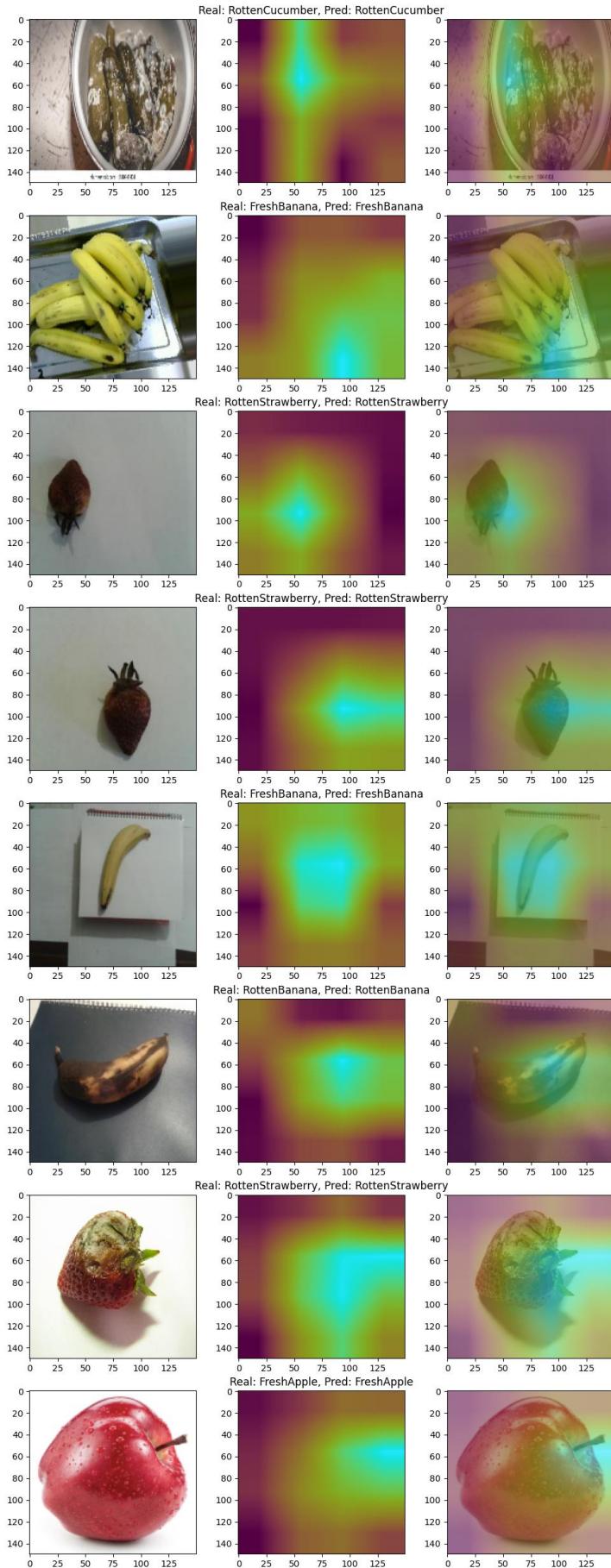
5.2. VGG16

Podstawowa sieć

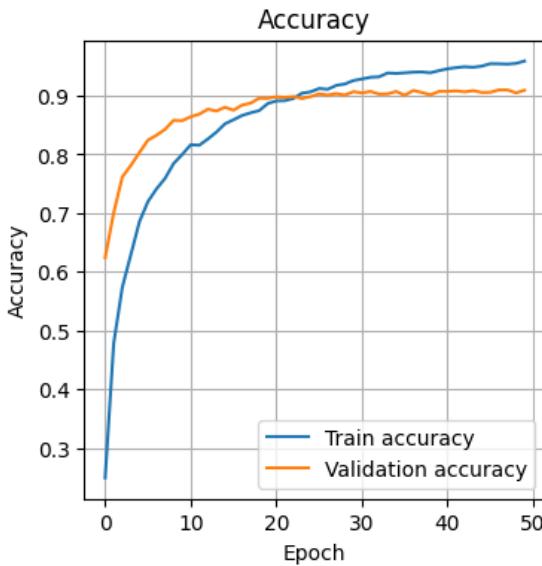
Dla podstawowej sieci VGG16 uzyskaliśmy dokładność wynoszącą 97.6%.



Rysunek 18. Macierz pomylek



Rysunek 19. Analiza GradCAM

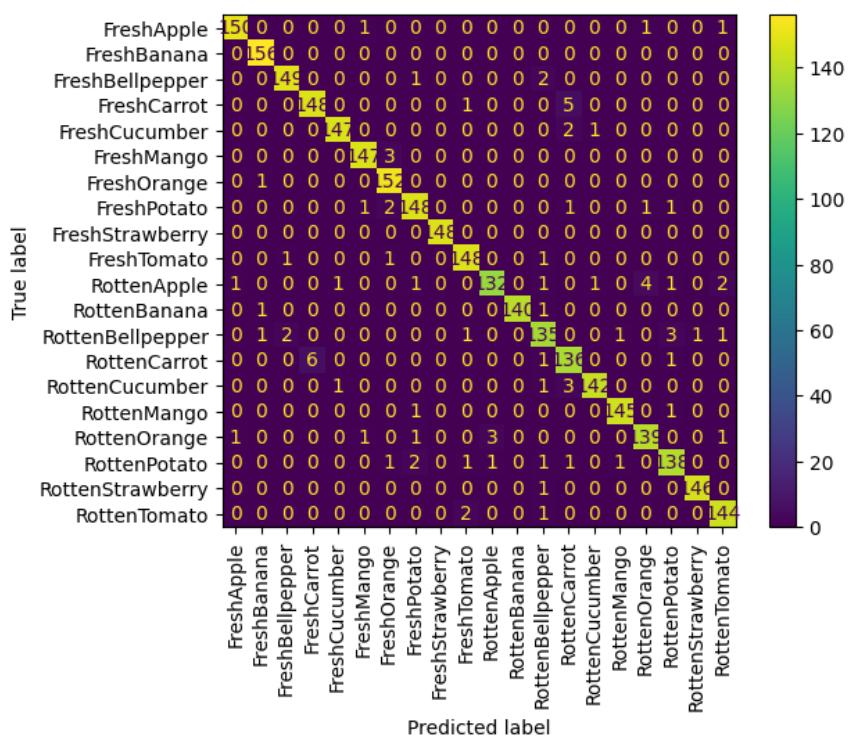


Rysunek 20. Dokładność w kolejnych epokach

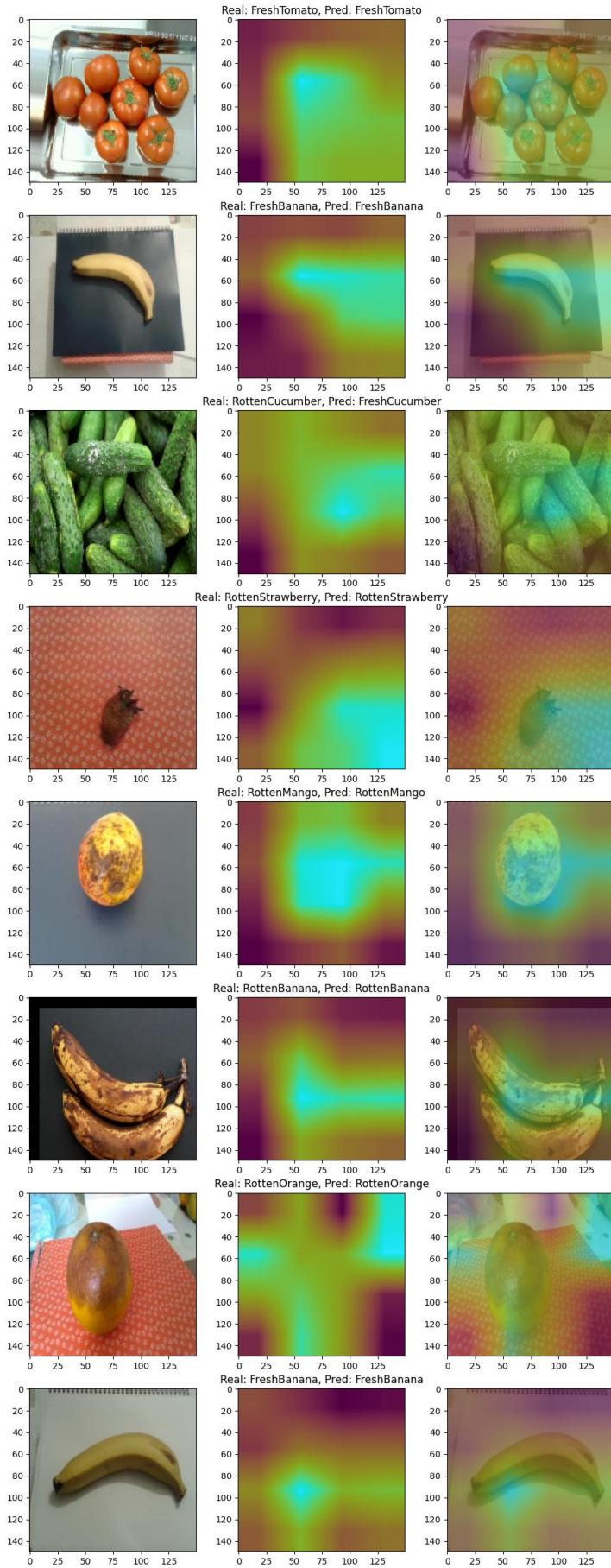
Podstawowa sieć działa na bardzo wysokiej dokładności, w macierzy pomyłek bardzo mało jest miejsc, które świadcząłyby o niepoprawnej klasyfikacji. Potwierdza to zarówno wyliczona wartość dokładności jak i wykres tychże parametrów otrzymanych przy uczeniu sieci. Widać na nim także, że wraz ze wzrostem liczby epok wzrasta jakość sieci i klasyfikacji obiektów. Analiza CAM pokazuje, że obszary skupienia znajdują się w miejscach najbardziej znaczących niezależnie od położenia lub ilości obiektów.

Odmrożona jedna warstwa

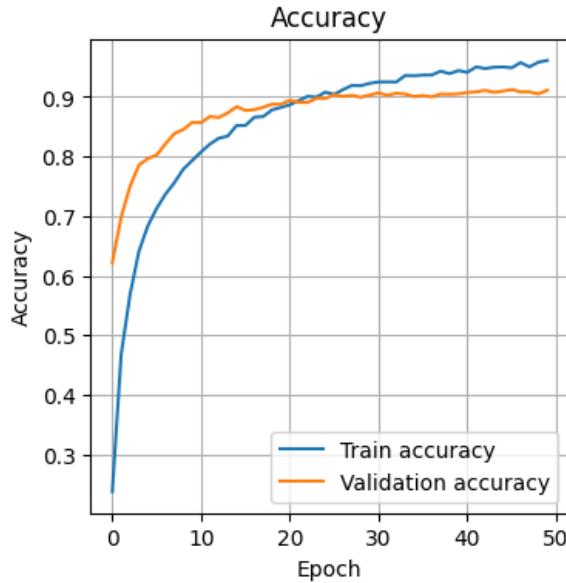
Dla sieci VGG16 z odmrożoną jedną warstwą otrzymaliśmy wynik dokładności na poziomie równym 97%.



Rysunek 21. Macierz pomyłek



Rysunek 22. Analiza GradCAM

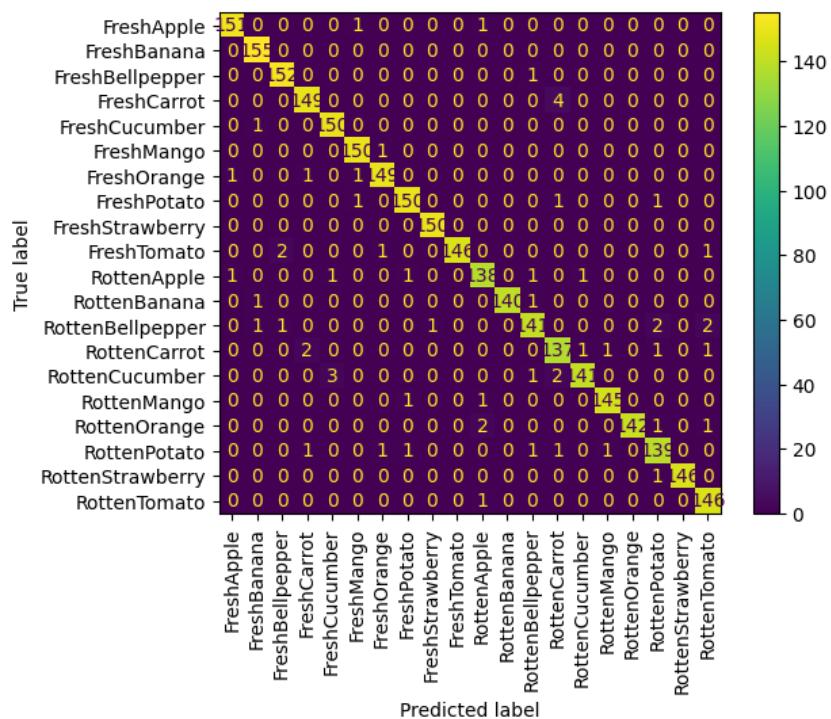


Rysunek 23. Dokładność w kolejnych epokach

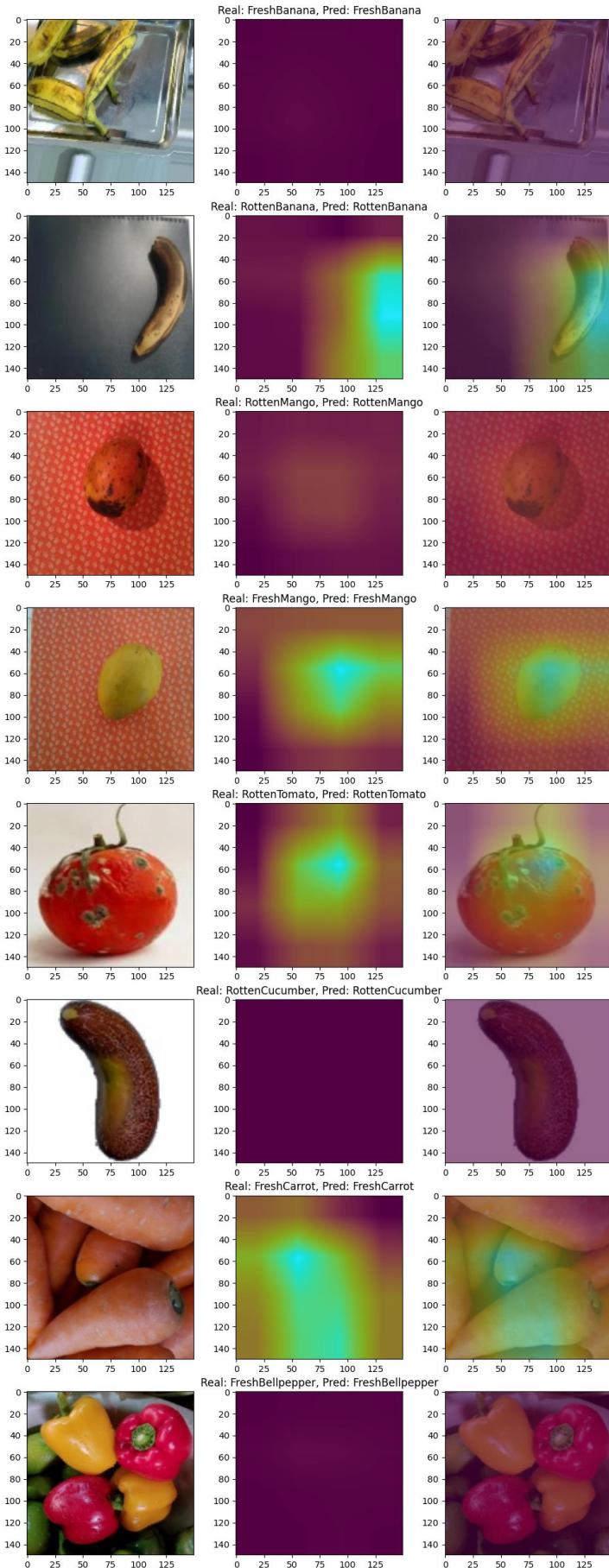
Porównując zachowanie podstawowej sieci oraz sieci z jedną odmrożoną ostatnią warstwą, ciężko zauważać znaczące różnice. Poziom dokładności jest bardzo podobny, w obu przypadkach wysoki. Wykres przedstawiający tą wartość otrzymaną na etapie trenowania również przebiega w prawie identyczny sposób. Analiza CAM również wskazuje na najbardziej znaczące obszary, co jest zachowaniem jak najbardziej poprawnym. Dla odnotowania warto jednak powiedzieć, że poziom dokładności jest odrobinę niższy.

Odmrożone dwie warstwy

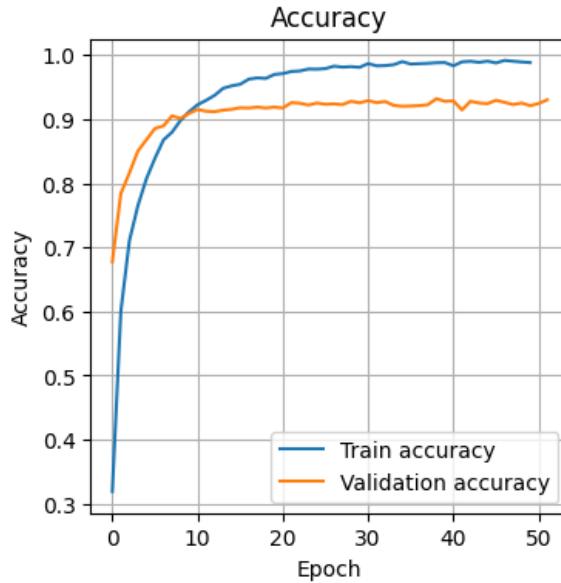
Dla sieci VGG16 z odmrożonymi dwiema warstwami otrzymaliśmy wynik dokładności na poziomie równym 98%.



Rysunek 24. Macierz pomylek



Rysunek 25. Analiza GradCAM

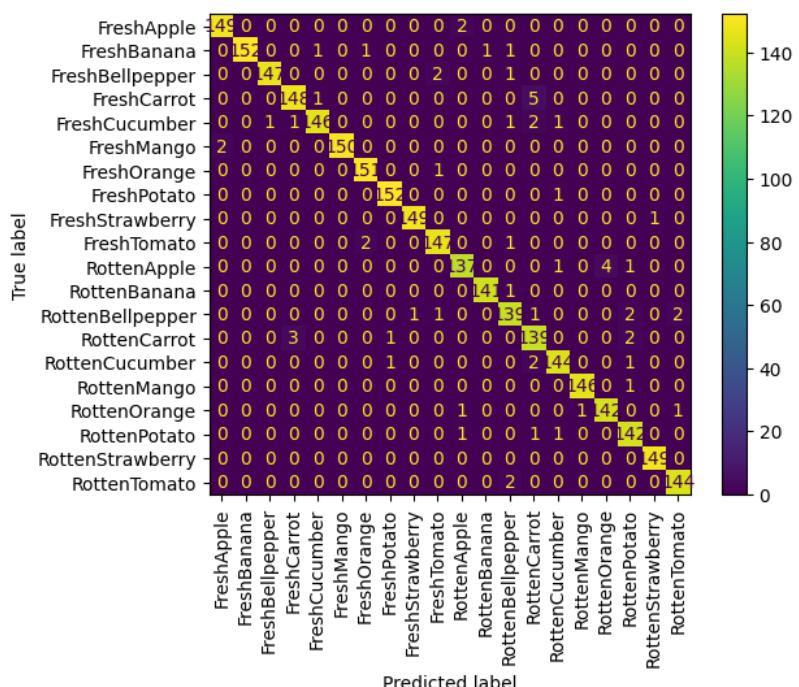


Rysunek 26. Dokładność w kolejnych epokach

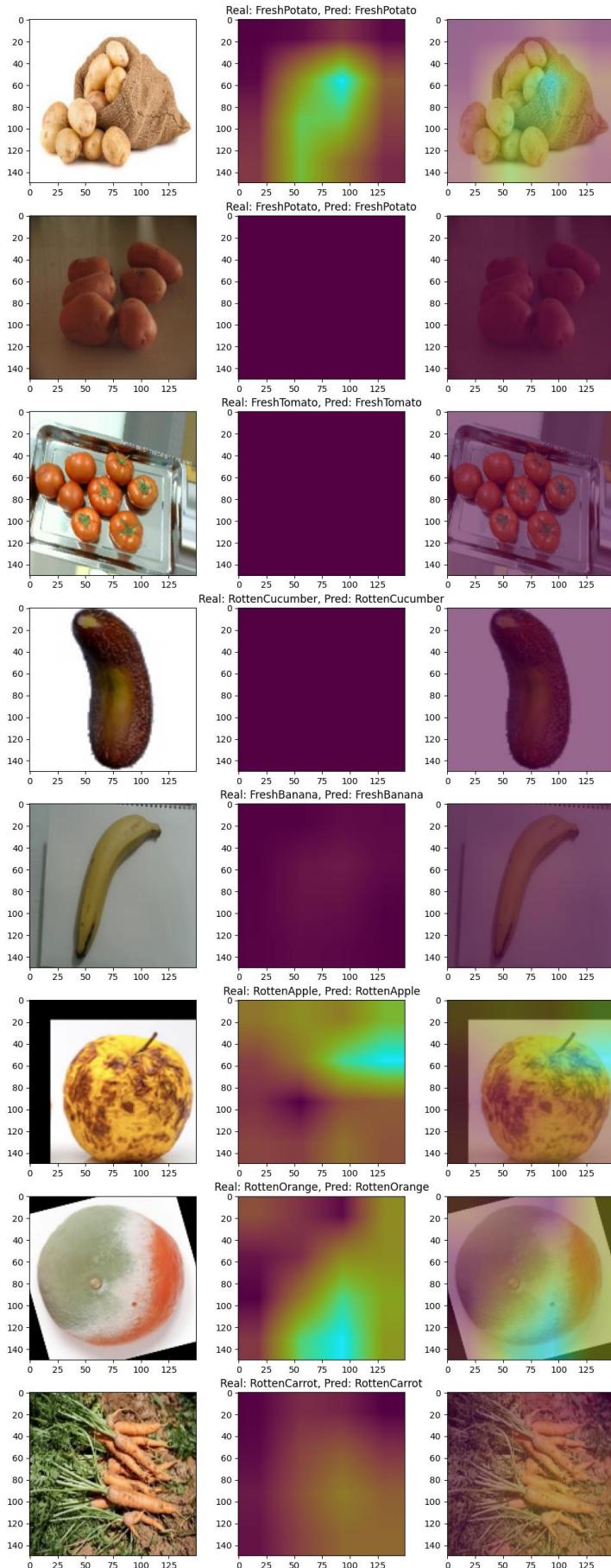
Tym razem, sieć z dwiema odmrożonymi warstwami, w przeciwieństwie do jednej odmrożonej, dała poprawę rezultatów w porównaniu z siecią podstawową. Dokładność, co prawda nieznacznie, ale jednak jest na wyższym poziomie. Widać to również na wykresie dokładności w zależności od ilości epok. Negatywne różnice widoczne są w analizie CAM, gdzie dla niektórych obiektów nie wykonały się lub wykonały, ale w bardzo mało znaczącym stopniu, heatmaps.

Odmrożone trzy warstwy

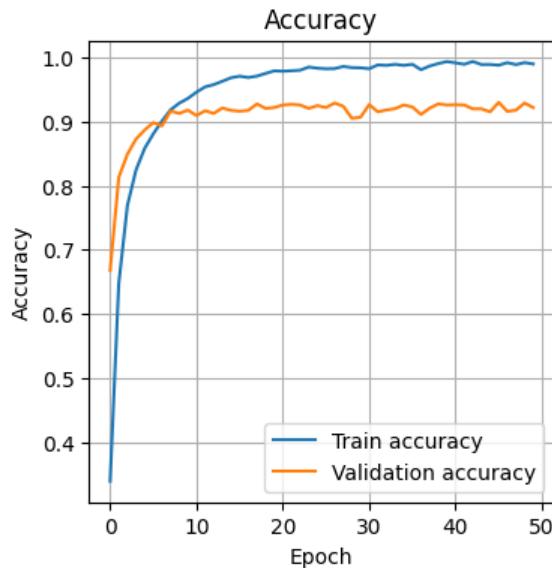
Dla sieci VGG16 z odmrożonymi dwiema warstwami otrzymaliśmy wynik dokładności na poziomie równym 97.9%.



Rysunek 27. Macierz pomyłek



Rysunek 28. Analiza GradCAM

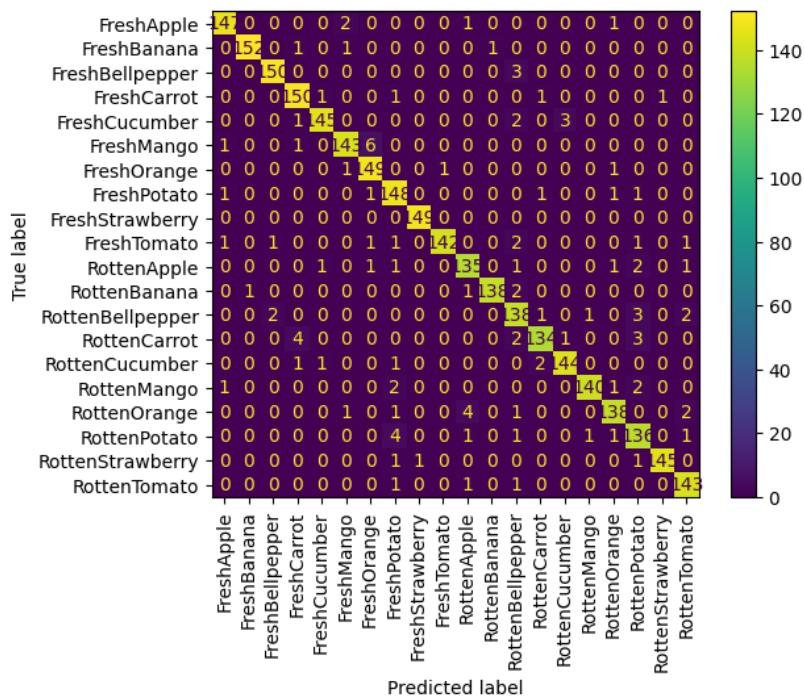


Rysunek 29. Dokładność w kolejnych epokach

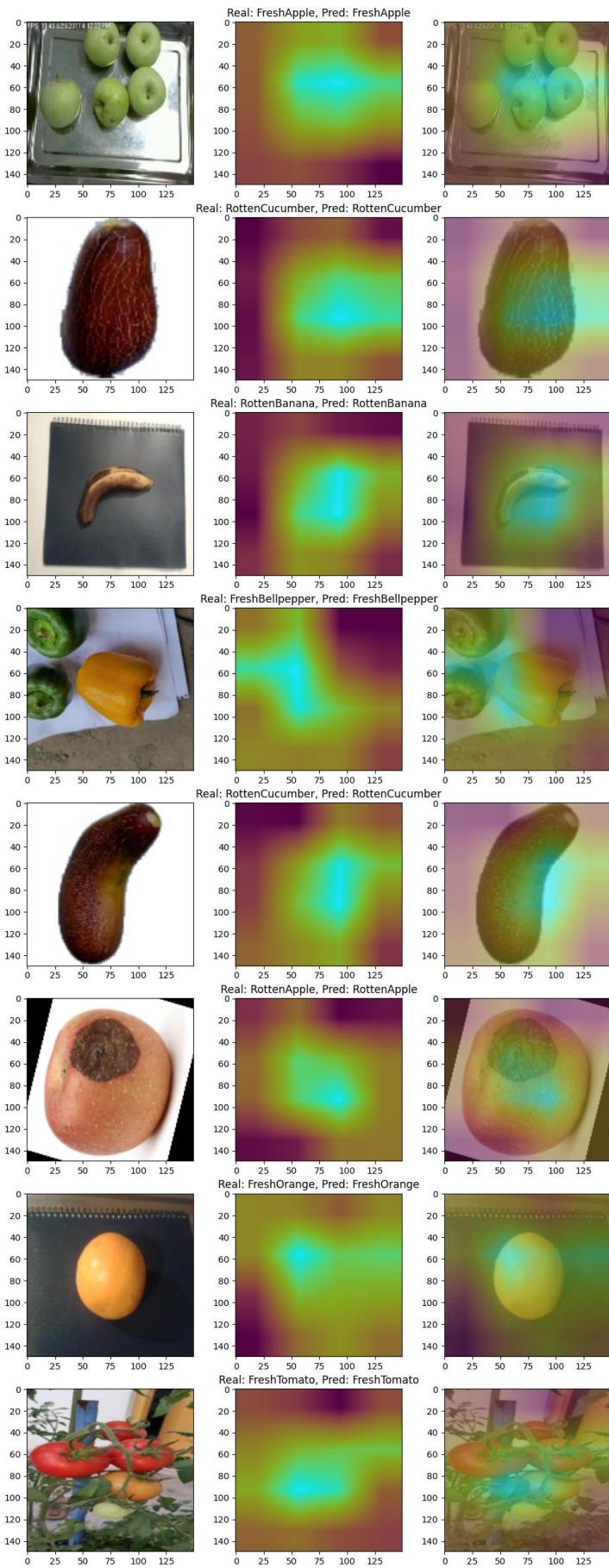
Model wytrenowany na sieci z trzema odmrożonymi warstwami zachowuje się bardzo podobnie do tego z dwoma odmrożonymi. Jego dokładność jest nieznacznie mniejsza, natomiast dalej na bardzo wysokim poziomie, o czym świadczy zarówno macierz pomyłek jak i wykres dokładności utworzony na podstawie wartości otrzymanych w trakcie trenowania. Również analiza CAM wygląda podobnie, ponieważ heatmapy albo się nie tworzą, albo nie są tak jednoznaczne w swoich obszarach.

Modyfikacja warstw własnych

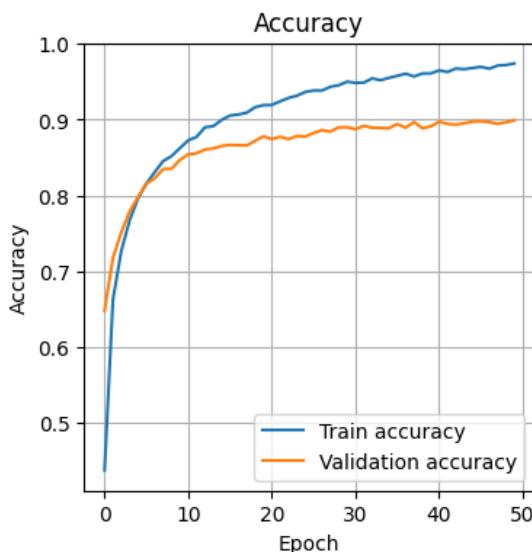
Dla sieci VGG16 ze zmodyfikowanymi warstwami, a konkretniej zawierający mniej warstw własnych (parametr *dense* ustawiony na wartość 20) otrzymaliśmy dokładność na poziomie 96%.



Rysunek 30. Macierz pomyłek



Rysunek 31. Analiza GradCAM



Rysunek 32. Dokładność w kolejnych epokach

Sieć z tak zmodyfikowanymi wartościami osiąga bardzo dobre rezultaty, co potwierdza macierz pomyłek, w której widać że zdecydowana większość obiektów jest poprawnie klasyfikowana. Na podstawie analizy CAM można stwierdzić, że brany pod uwagę jest obszar znaczący danego warzywa lub owocu niezależnie od tego, gdzie i w jakiej ilości się on znajduje. Wykres dokładności zawierający wartości otrzymane przy trenowaniu modelu potwierdza wysoką wydajność sieci. Czas trenowania wynoszący około półtorej godziny ma więc tutaj uzasadnienie.

5.3. Porównanie czasu uczenia sieci

Dla obu sieci czas trwania uczenia jednej epoki wahał się od ok. 90 do 170s.

```

Epoch 6/50
262/262 [=====] - 171s 651ms/step
Epoch 7/50
262/262 [=====] - 168s 641ms/step
Epoch 8/50
262/262 [=====] - 171s 653ms/step
Epoch 9/50
262/262 [=====] - 175s 667ms/step

```

Rysunek 33. Czas dla bazowej sieci ResNet50

```

Epoch 8/50
262/262 [=====] - 89s 338ms/step
Epoch 9/50
262/262 [=====] - 87s 333ms/step
Epoch 10/50
262/262 [=====] - 87s 331ms/step
Epoch 11/50
262/262 [=====] - 88s 334ms/step

```

Rysunek 34. Czas dla ResNet50 z dodatkowymi warstwami

```

Epoch 6/50
280/280 [=====] - 114s 407ms/step
Epoch 7/50
280/280 [=====] - 110s 394ms/step
Epoch 8/50
280/280 [=====] - 110s 395ms/step
Epoch 9/50
280/280 [=====] - 113s 405ms/step

```

Rysunek 35. Czas dla VGG16 z jedną odmrożoną warstwą

```

Epoch 10/50
280/280 [=====] - 117s 418ms/step
Epoch 11/50
280/280 [=====] - 115s 410ms/step
Epoch 12/50
280/280 [=====] - 119s 423ms/step
Epoch 13/50
280/280 [=====] - 121s 431ms/step

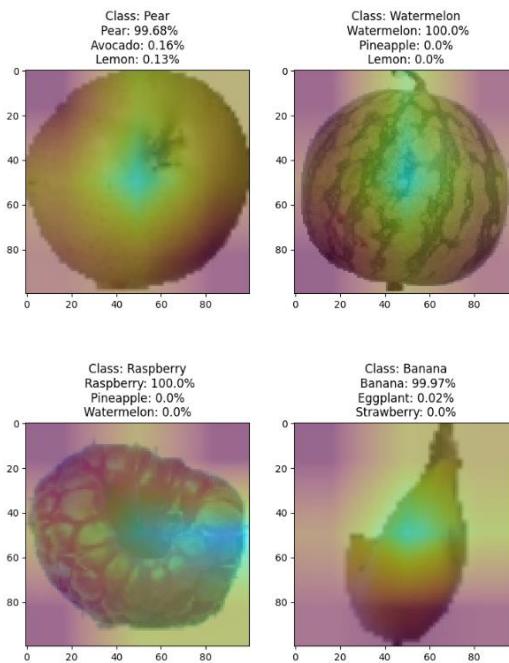
```

Rysunek 36. Czas dla VGG16 z dwiema odmrożonymi warstwami

Cieźko jest jednak wskazać, dla której sieci lub przypadku czas był krótszy. Jest on natomiast podobny dla testów wykonywanych przez tę samą osobę. Można więc wysnuć wniosek, że zależy on od parametrów środowiska wykonawczego, a niekoniecznie od użytej sieci lub jej parametrów. Powodem braku tej zależności może być fakt, że wykorzystujemy technikę transfer learningu, a więc w każdej epoce uczone jest tylko kilka ostatnich warstw, a nie cała sieć.

5.4. Poprzedni dataset

Początkowo planowaliśmy zastosować naszym projekcie inny dostępny dataset [2]. Zawiera on zdjęcia owoców na białym tle. Jednak po zastosowaniu dla niego analizy GradCAM zauważliśmy, że sieć bierze zawsze pod uwagę środek obrazu, ponieważ to właśnie tam znajduje się obiekt. Wyniki nie były wystarczająco interesujące, zdecydowaliśmy się więc na wybranie innego datasetu.



Rysunek 37. Analiza GradCAM dla poprzedniego datasetu

6. Wnioski

Zarówno sieć ResNet50 jak i VGG16 w swojej podstawowej wersji uzyskują wysoką dokładność dla badanego problemu. Różnice występują jednak dla wprowadzanych przez nas zmian, np. odmrażania warstw. Zdecydowanie lepsze wyniki uzyskuje wtedy sieć VGG, której dokładność utrzymuje się na poziomie powyżej 95%. W przypadku odmrażania warstw dla sieci ResNet dokładność spada nawet do 50%, co jest bardzo złym wynikiem. Sieć VGG jest więc prostsza do zastosowania w naszym problemie, ponieważ w jej przypadku występuje mniejsze ryzyko drastycznego spadku dokładności.

Przykład sieci ResNet pokazuje na trudności jakie można napotkać w przypadku prób samodzielnego tworzenia lub nawet niewielkiego zmieniania sieci. Biorąc pod uwagę długi czas uczenia sieci, trudności te mogą okazać się dużym problemem. Wskazuje to na dużą rolę transfer learningu i wykorzystania już istniejących rozwiązań. Pozwala to zaoszczędzić bardzo dużo czasu poświęcanego na znalezienie odpowiednich parametrów sieci i otrzymanie w dość prosty sposób rozwiązań charakteryzujących się wysoką dokładnością.

7. Literatura

- [1] <https://www.kaggle.com/datasets/muhriddinmuxiddinov/fruits-and-vegetables-dataset>
- [2] <https://www.kaggle.com/datasets/moltean/fruits>

8. Podział pracy

	Patryk Chorąży	Rafał Kośla	Artur Mzyk	Joanna Nużka	Adrian Poniatowski	Wojciech Poniewierka
Dataset			100%			
Sieć VGG		30%			30%	40%
Sieć ResNet	30%		40%	30%		
Analiza CAM			30%			70%
GUI		100%				
Eksperymenty	25%			25%	50%	
Dokumentacja	20%		20%	40%	20%	