

1. Introduction

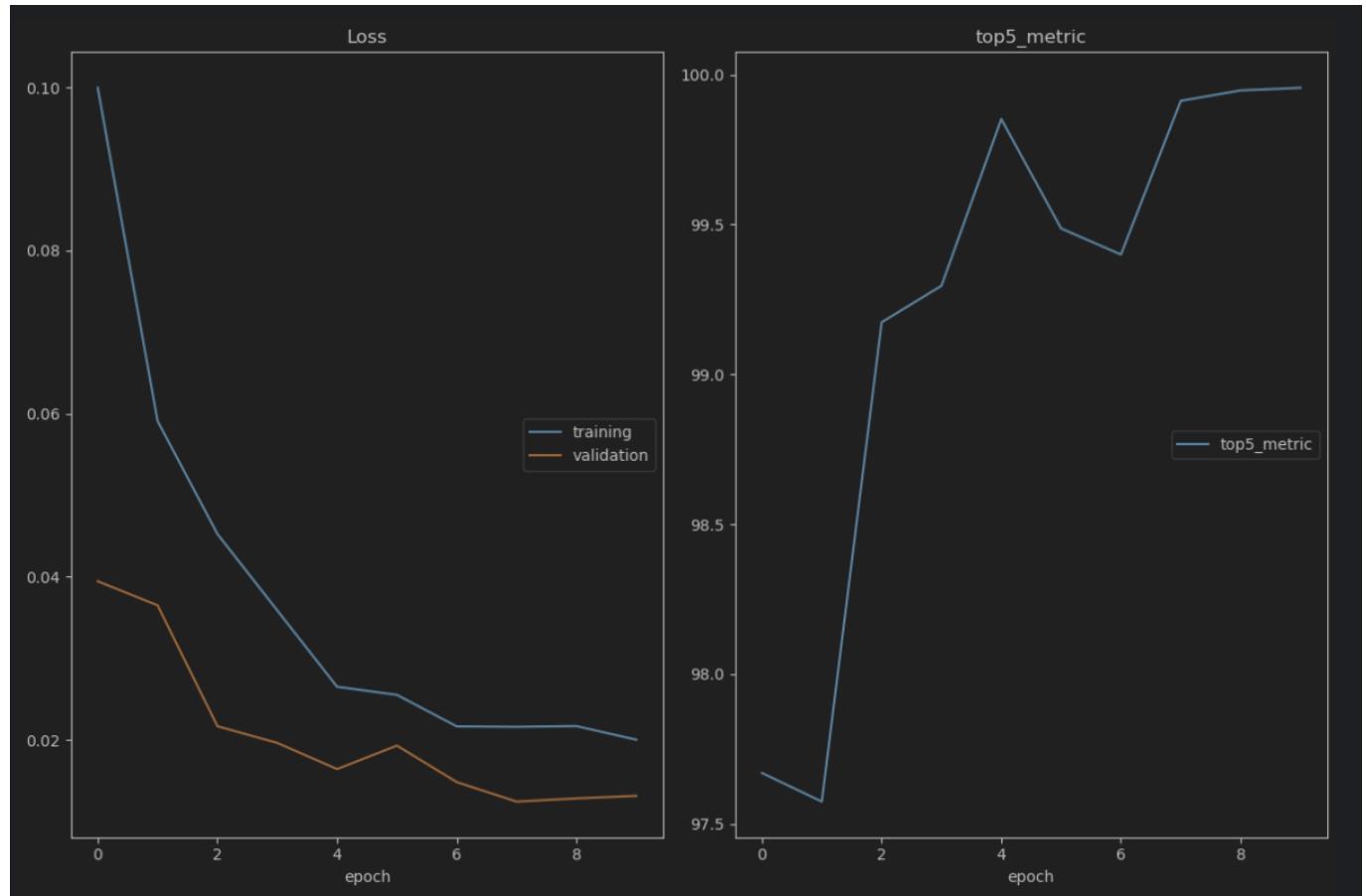
1.1. Fruits360

Fruits360 dataset

Contrastive loss

Euclidean Distance

1.1.1. test1



Rysunek 1: Fruits360

175 przykładów na epokę.

10 epok

batchSize : 16

obrazy wejściowe , są resize() do rozmiarów 100x100.

TOP5 metryka = 99.95%

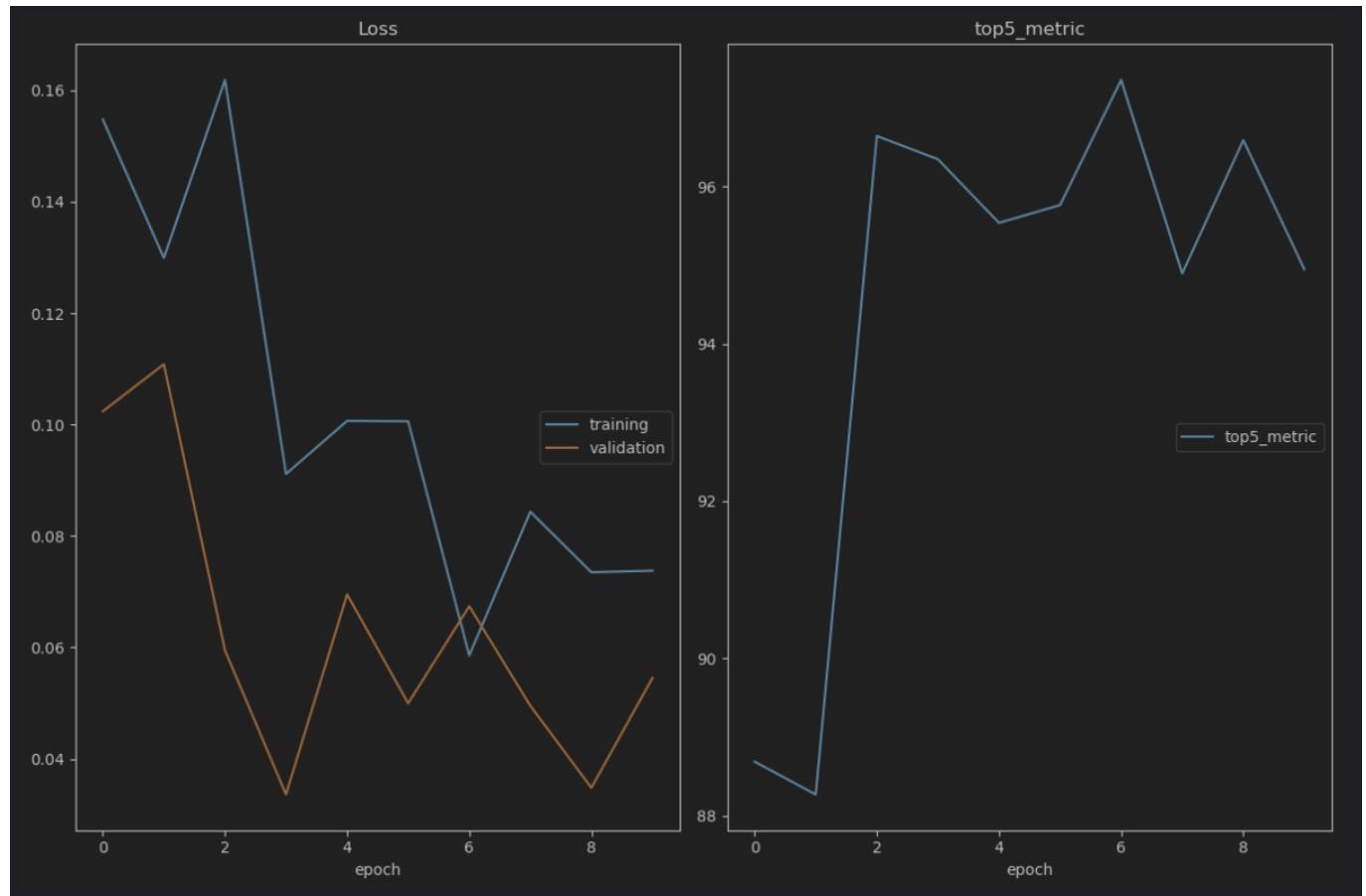
1.1.2. test2

Fruits360 dataset

Contrastive loss

Euclidean Distance

18 przykładów na epokę.



Rysunek 2: Fruits 360

obrazy wejściowe są resize() do 100x100

Top5 metryka = 95.11%

1.2. FruitsRecognition

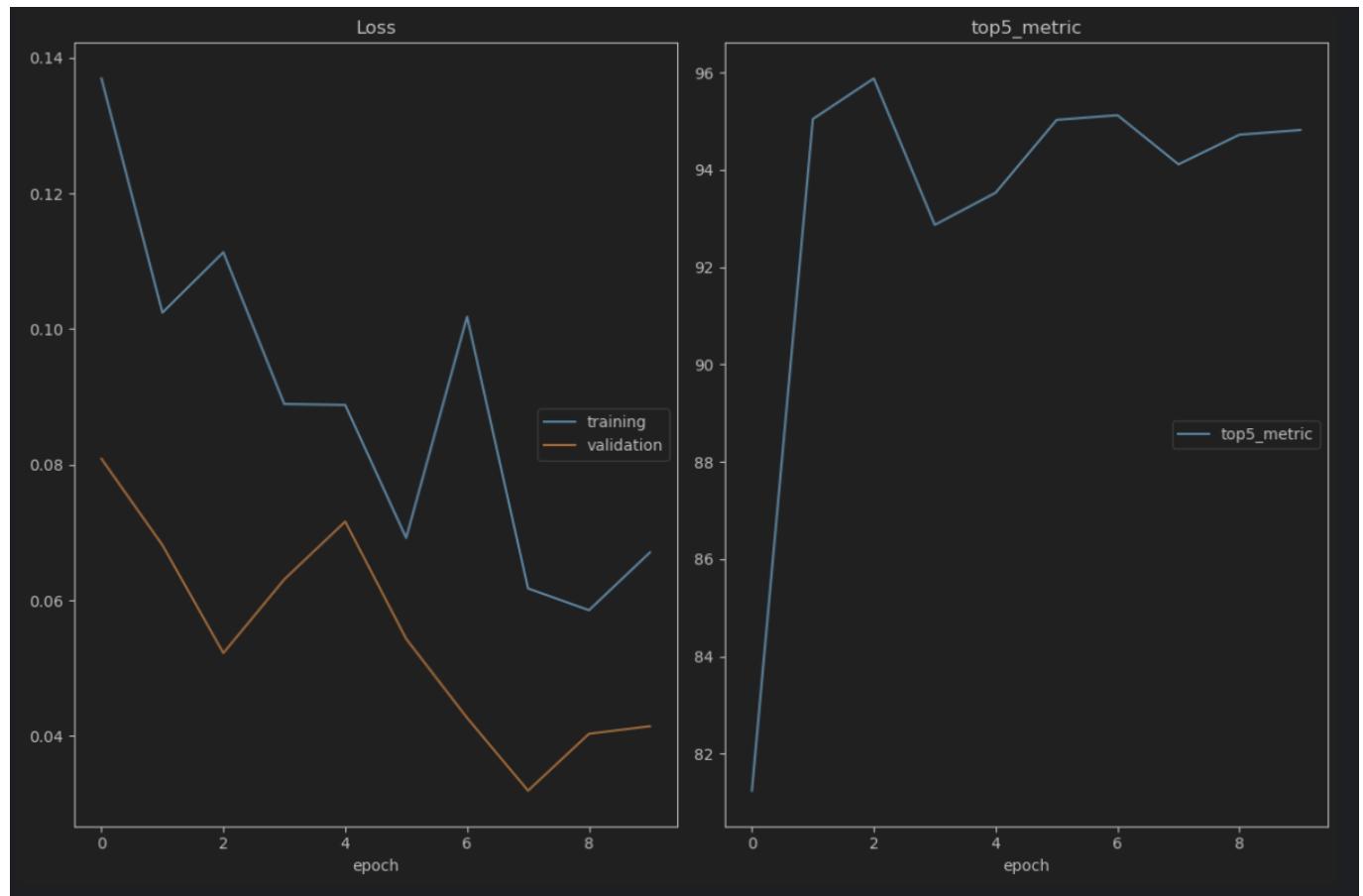
1.2.1. test3

Fruits360 dataset

Contrastive loss

Euclidean Distance

18 przykładów na epokę.



Rysunek 3: FruitsRecognition

obrazy wejściowe, resize() do 250x250.

Top5 metryka = 94.81%

2. FRUITS RECOGNITION DATASET

2.1. test1

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive Loss

Obliczna odległość : Euclidean Distance

Wartości: reshape() obrazów do 250x250

ilość zdjęć na epokę: 175

ilość epok : 10

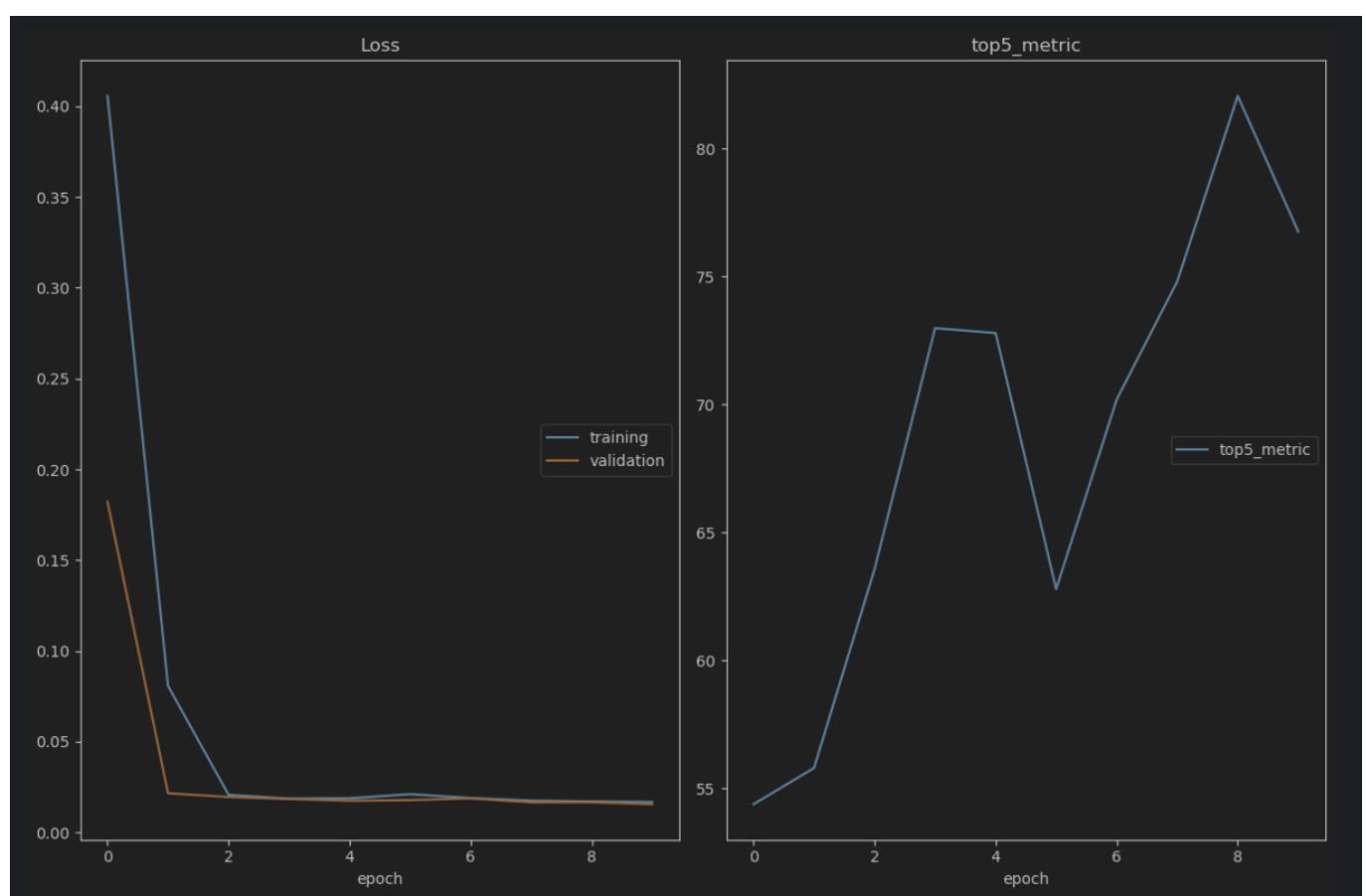
batchSize : 16

Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : **0.3**

Model BEZ Transfer Learning.

WYNIKI funkcji strat i top5 metryki po każdej epoce:



Rysunek 4: Loss i top5 metryka

2.2. test2

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive loss

Wartości: reshape() obrazów do 250x250

ilość zdjęć na epoce: 175

ilość epok : 10

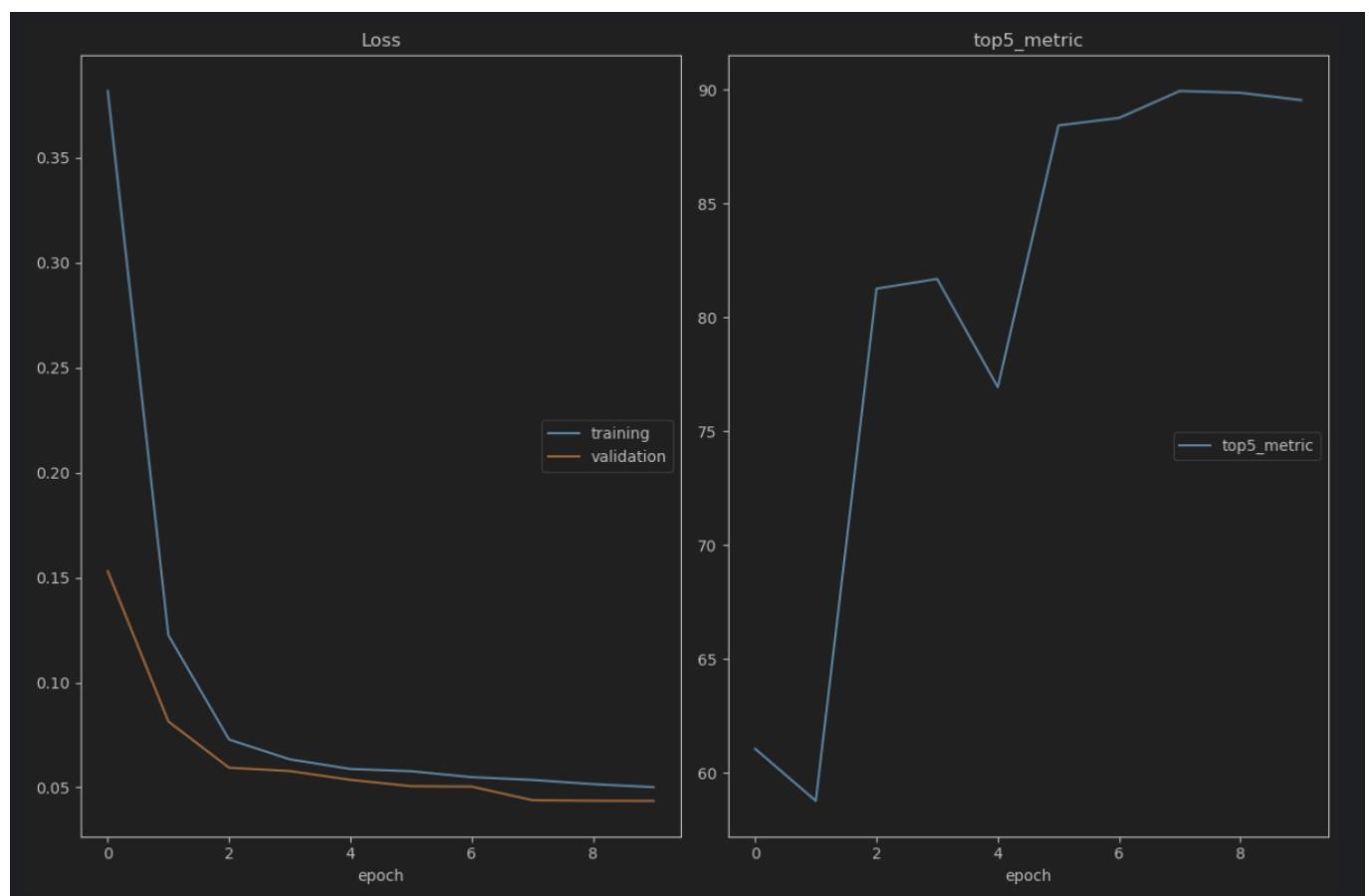
batchSize : 16

Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : **0.6**

Model BEZ Transfer Learning.

WYniki funkcji strat i top5 metryki po każdej epoce:



Rysunek 5: Loss i top5 metryka

2.3. test2

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive loss

Wartości: reshape() obrazów do 224x224 (takie wejścia są wymagane w modelu VGG16)

ilość zdjęć na epokę: 175

ilość epok : 10

batchSize : 16

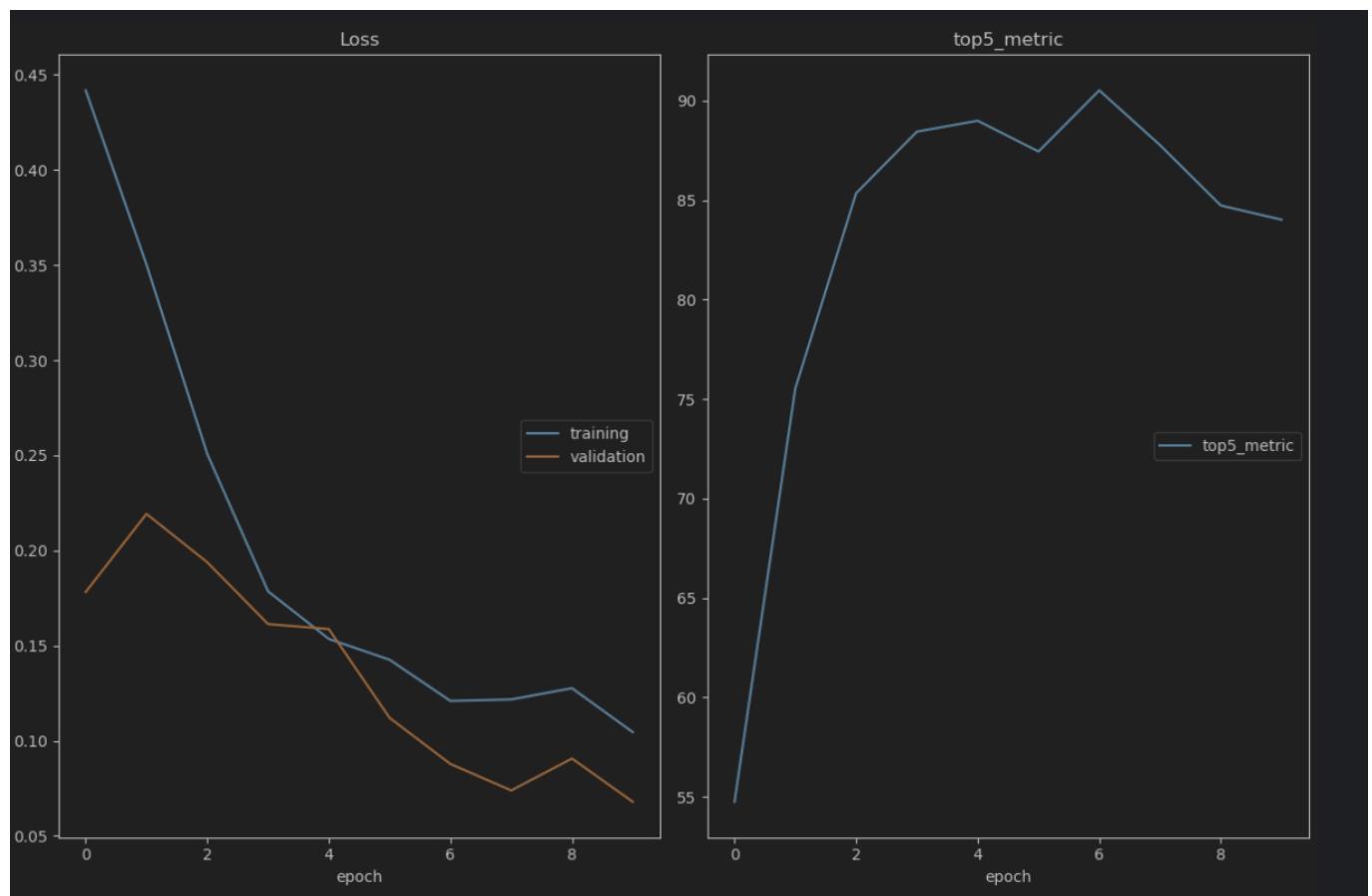
Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : 0.6

TRANSFER LEARNING.

model VGG16. Uczenie warstw gestyckich. Dense64 + Dense16

1,600,000 parametrów trenowalnych WYniki funkcji strat i top5 metryki po każdej epoce:



Rysunek 6: Loss i top5 metryka

Rysunek 7: Loss i top5 metryka

2.4. test2

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive loss

Wartości: reshape() obrazów do 250x250

ilość zdjęć na epokę: 175

ilość epok : 10

batchSize : 16

Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : 0.6

TRANSFER LEARNING.

model VGG16 + Dense64+ Dense16

FINE TUNING 3 ostatnich warstw konwolucyjnych.

WYniki funkcji strat i top5 metryki po każdej epoce:

Problem z załadanowaniem wyuczonego modelu :

zwracany błąd: ValueError: axes don't match array

Mimo iż nie zmieniałem struktury.

Pamiętam że raz na początku mi zadziałało. Dlatego muszę poszukać czego to jest kwestia.

2.5. test2

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive loss

Wartości: reshape() obrazów do 224x224 (takie wejścia są wymagane w modelu VGG16)

ilość zdjęć na epokę: 600

ilość epok : 6

batchSize : 16

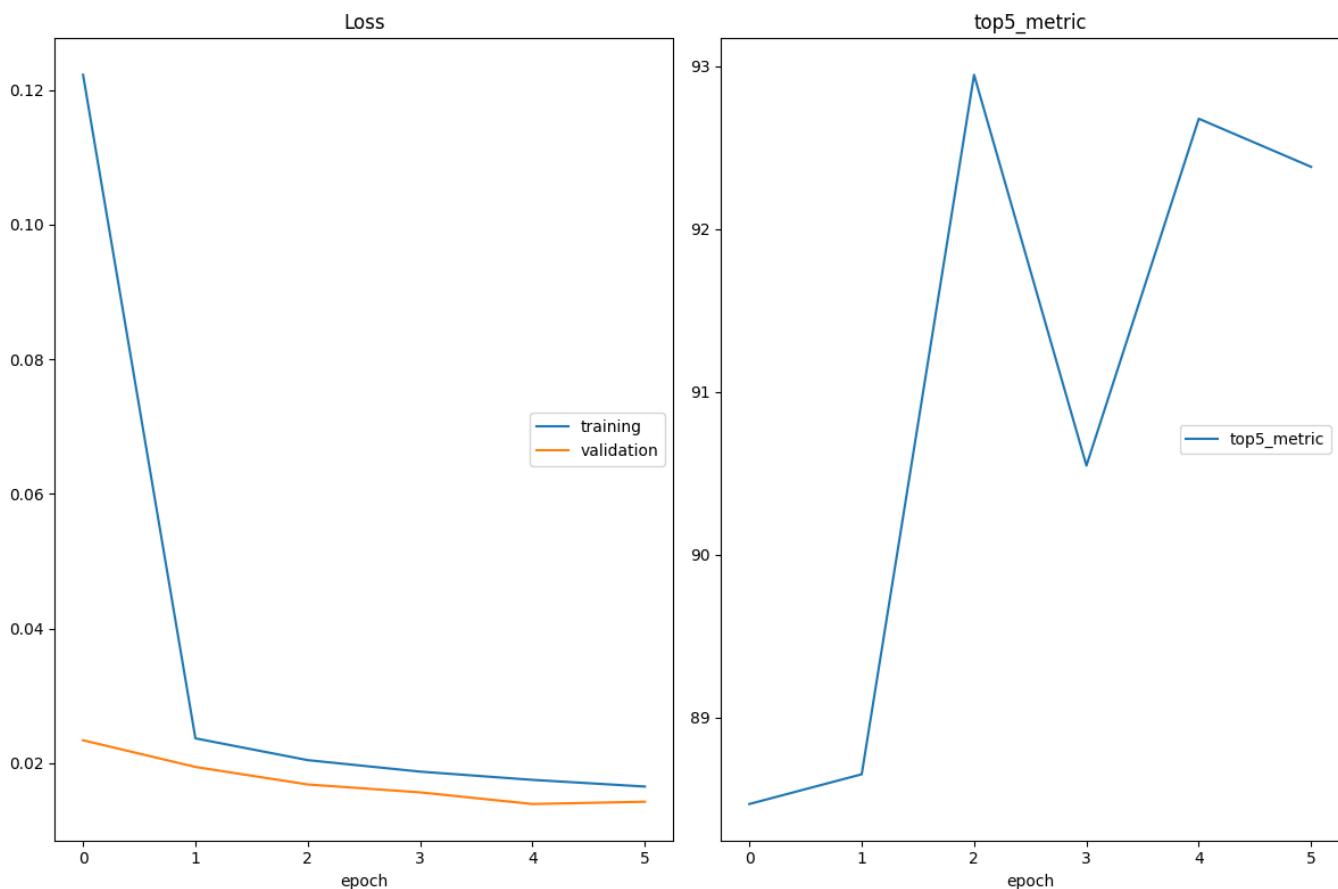
Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : 0.6

TRANSFER LEARNING.

model VGG16. Uczenie warstw gestyckich. Dense512 + Dense16

12 854 800 parametrów trenowalnych WYniki funkcji strat i top5 metryki po każdej epoce:



Rysunek 8: Loss i top5 metryka

2.6. test2

Test prowadzony dla danych : **Fruits Recognition**.

Contrastive loss

Wartości: reshape() obrazów do 224x224 (takie wejścia są wymagane w modelu VGG16)

ilość zdjęć na epokę: 600

ilość epok : 6

batchSize : 16

starting learning rate : ????

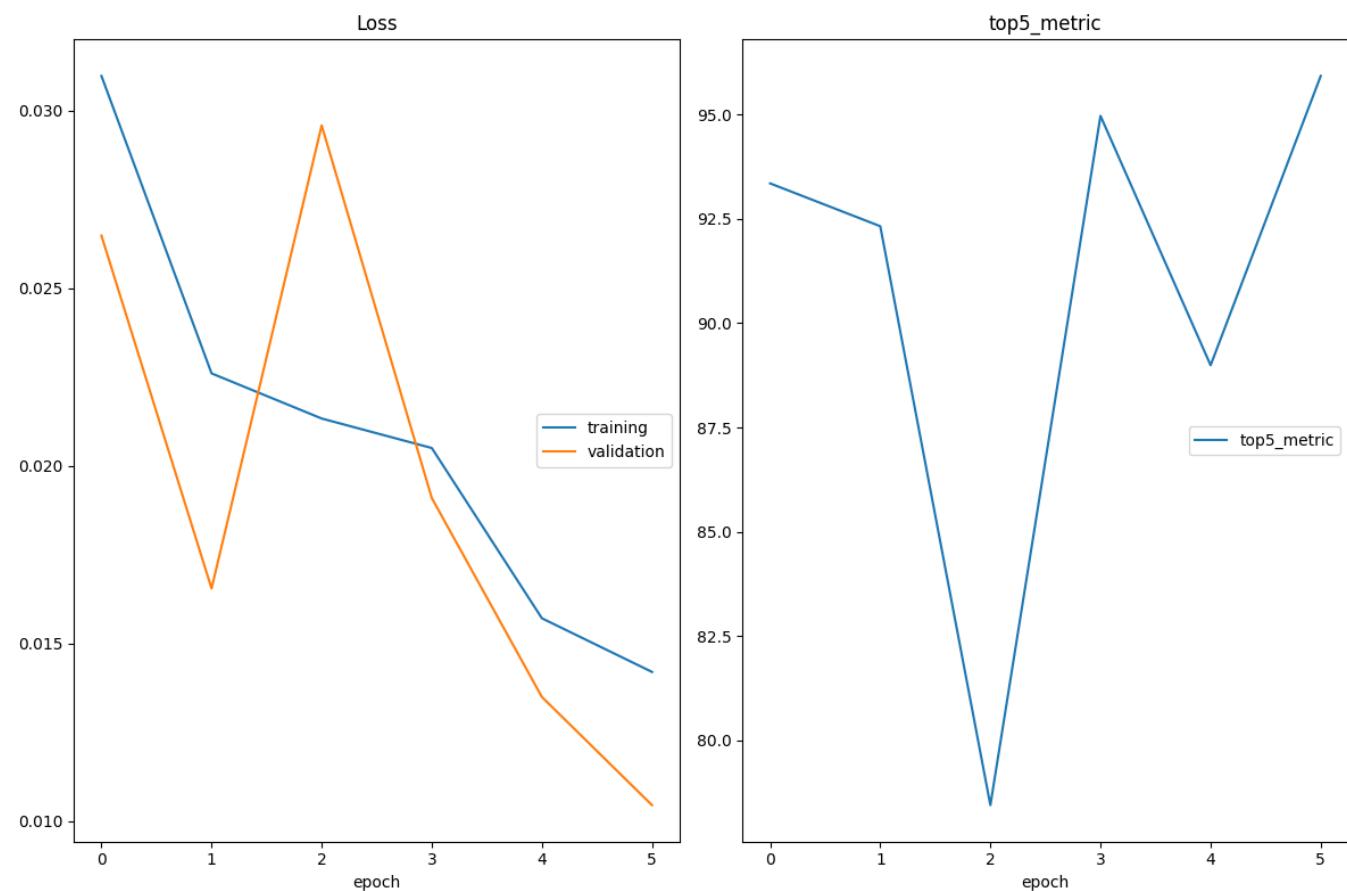
Liczba reprezentantów : 5 (dobrani według wytycznych)

margin : 0.6

TRANSFER LEARNING.

model VGG16+ Dense512 + Dense16

FINE TUNING 3 ostatnich warstw (optimizer = SGD) 17 854 800 parametrów trenowalnych WYNIKI funkcji strat i top5 metryki po każdej epoce:



Rysunek 9: Loss i top5 metryka

2.7. test2

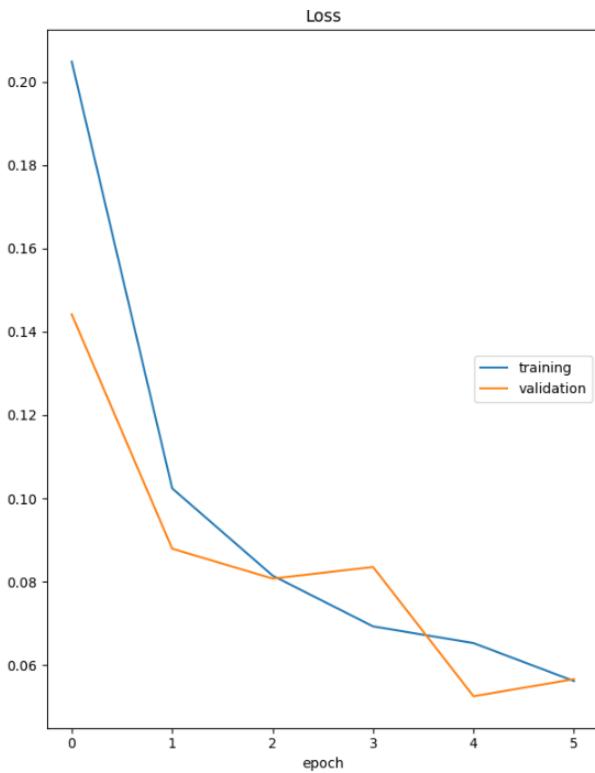
#dims	VAL
64	86.8% \pm 1.7
128	87.9% \pm 1.9
256	87.7% \pm 1.9
512	85.6% \pm 2.0

Table 5. Embedding Dimensionality. This Table compares the effect of the embedding dimensionality of our model NN1 on our hold-out set from section 4.1. In addition to the VAL at 10^{-3} we also show the standard error of the mean computed across five splits.

Rysunek 10: From Document (can give us the meaning of scaling embedding)

2.8. Test Dense32 Dense16 Triplet

- *Metoda* : Triplet loss
- *Wartsty* : Dense 32, Dense 16
- *Transfer learning* : Tak
- *Model bazowy* : VGG16
- *Training triplets* : 4800
- *Validation triplets* : 1200
- *Batch size* : 16
- *Epoki* : 6
- *Image size* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantw* : 5
- *Optimizer* : Adam 0.001



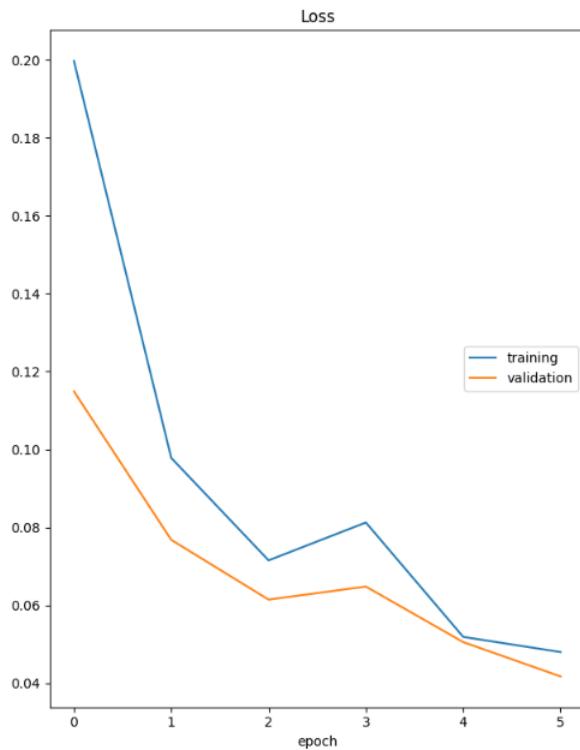
Rysunek 11: Before Finetuning

- *Skutecznosc modelu* : 0.1148
- *Ilosc danych testowych* : 6705

Po finetuningu z przetrenowanymi 3 warstwami początkowymi VGG16 oraz optimizерem SGD z learning rate 0.001, momentum 0.9, wartość training loss i validation loss oscylowały w okolicach 0.5.

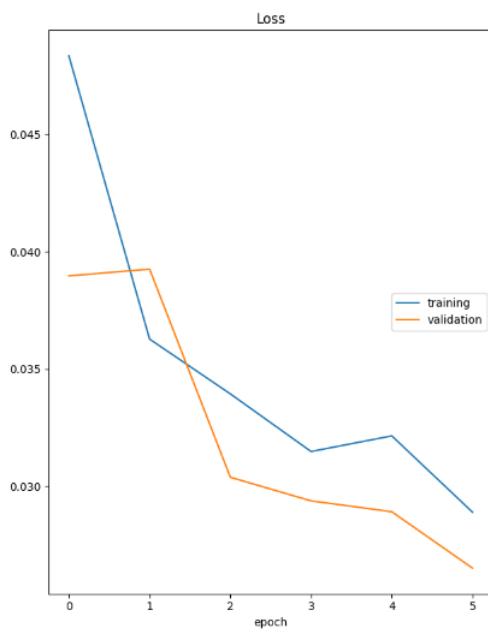
2.9. Test Dense512 Dense16 Triplet

- *Metoda* : Triplet loss
- *Wartsty* : Dense 512, Dense 16
- *Transfer learning* : Tak
- *Model bazowy* : VGG16
- *Training triplets* : 4800
- *Validation triplets* : 1200
- *Batch size* : 16
- *Epoki* : 6
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : Adam 0.0001



Rysunek 12: Before finetuning

- *Skutecznosc modelu* : 0.8530
- *Ilosc danych testowych* : 6705



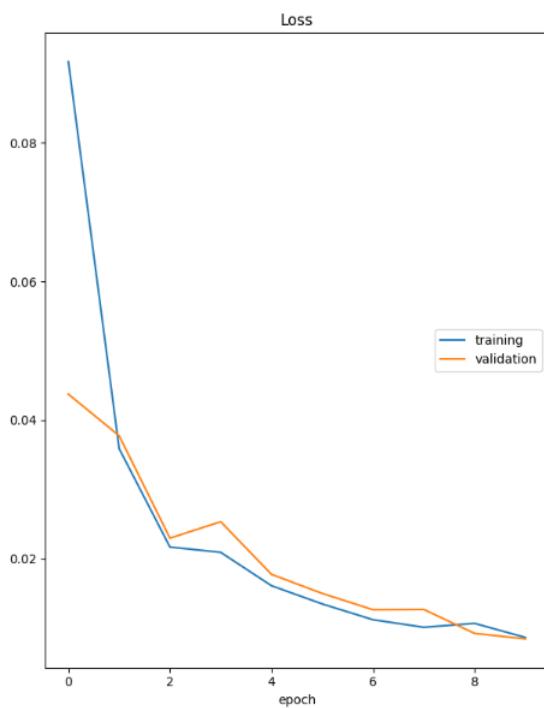
Rysunek 13: After finetuning

Finetuning:

- **Warstwy przetrenowane :** [-3:]
- **Optimizer :** SGD 0.0001
- **Skutecznosc modelu :** 0.8621
- **Ilosc danych testowych :** 6705

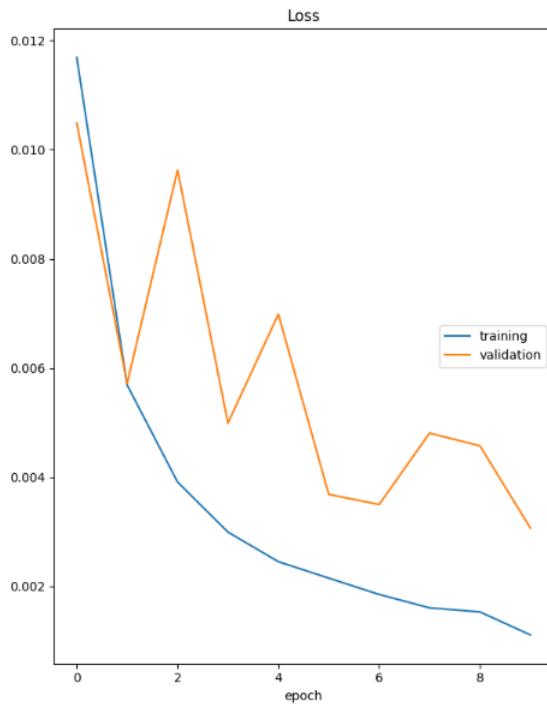
2.10. Test Dense512 Dense16 Triplet - Big data

- *Metoda* : Triplet loss
- *Wartsty* : Dense 512, Dense 16
- *Transfer learning* : Tak
- *Model bazowy* : VGG16
- *Training triplets* : 31000
- *Validation triplets* : 6700
- *Batch size* : 16
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : Adam 0.001



Rysunek 14: Before finetuning

- *Skutecznosc modelu(TOP5)* : 0.8577
- *Ilosc danych testowych* : 6705



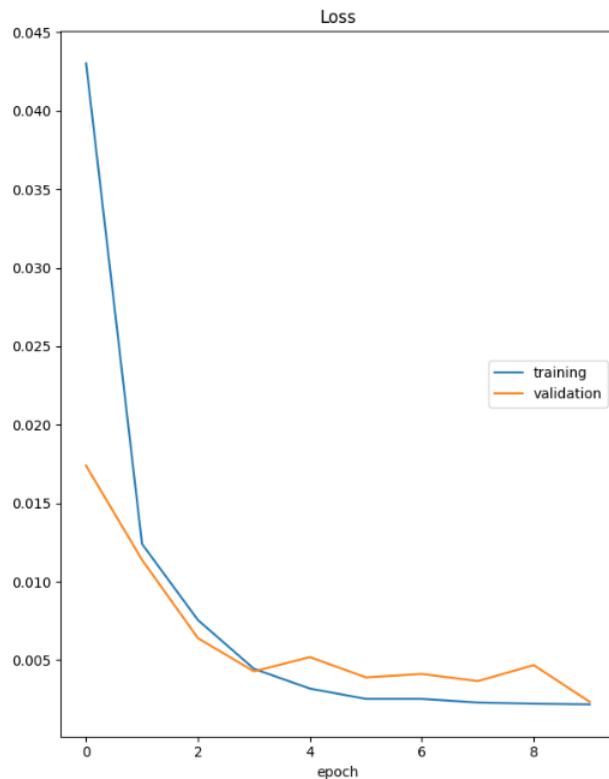
Rysunek 15: After finetuning

Finetuning:

- *Warstwy przetrenowane : [-4:]*
- *Optimizer : SGD 0.001*
- *Skutecznosc modelu(TOP5) : 0.9441*
- *Ilosc danych testowych : 6705*

2.11. Test Dense512 Dense256 Triplet - Big data

- *Metoda* : Triplet loss
- *Wartsty* : Dense 512, Dense 256
- *Transfer learning* : Tak
- *Model bazowy* : Xception
- *Training triplets* : 31000
- *Validation triplets* : 6700
- *Barch size* : 128
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : Adam 0.001

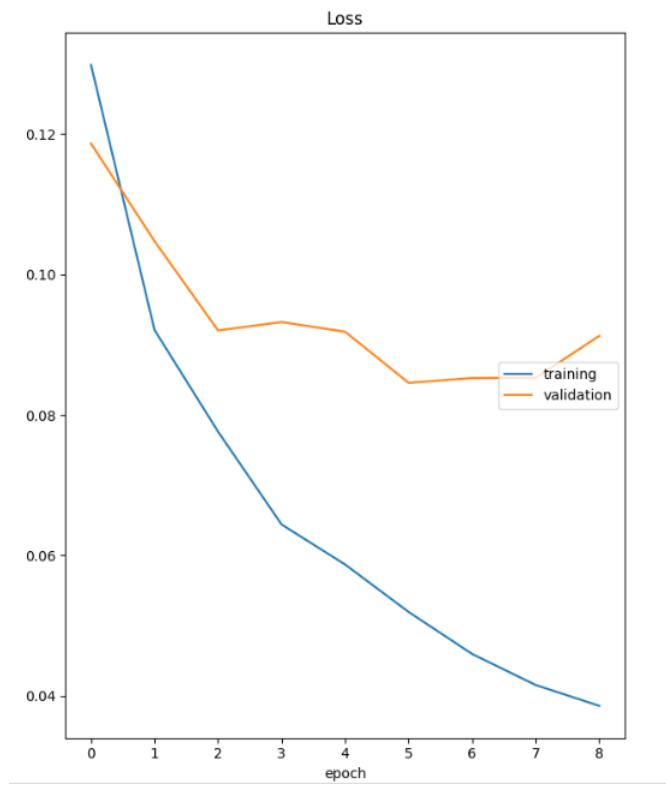


Rysunek 16: Before finetuning

- *Skutecznosc modelu(TOP5)* : 0.9993
- *Ilosc danych testowych* : 6705

2.12. Test Dense512 Dense256 Triplet - Veg dataset

- *Metoda* : Triplet loss
- *Wartsty* : Dense 512, Dense 256
- *Transfer learning* : Tak
- *Model bazowy* : Xception
- *Training triplets* : 72779
- *Validation triplets* : 8999
- *Barch size* : 128
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : Adam 0.001

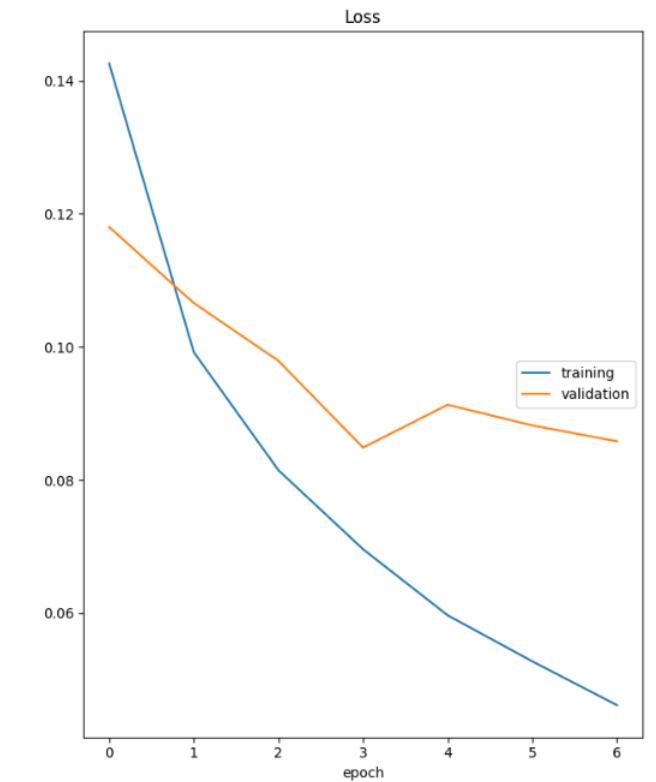


Rysunek 17: Before finetuning (Visible overfitting)

- *Zbior testowy* : FruitRecognition
- *Skutecznosc modelu(TOP5)* : 0.7840
- *Ilosc danych testowych* : 6705

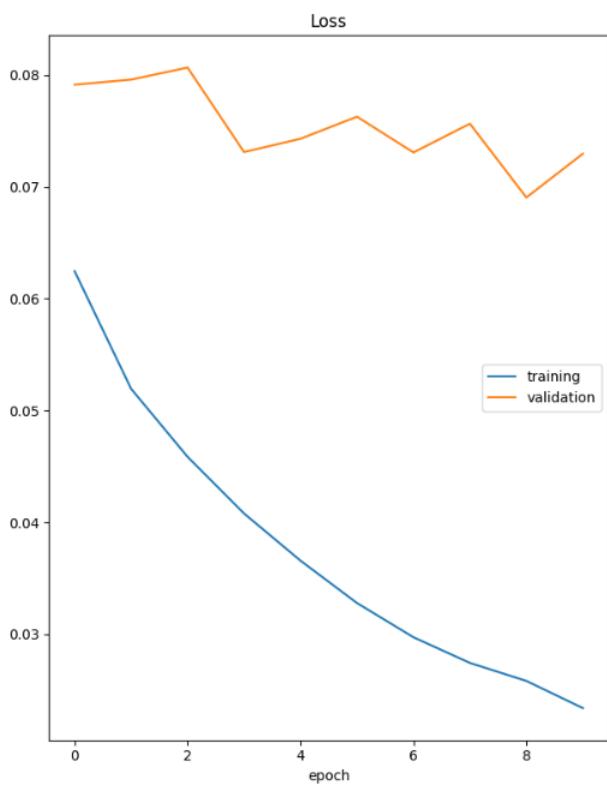
2.13. Test Dense512 Dense256 Triplet - VegFru dataset

- *Metoda* : Triplet loss
- *Wartsty* : Dense 512, Dense 256
- *Transfer learning* : Tak
- *Model bazowy* : Xception
- *Training triplets* : 72779 (Veg) + 55638 (Fru)
- *Validation triplets* : 8999 (Veg) + 6914 (Fru)
- *Batch size* : 128
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : Adam 0.001



Rysunek 18: Before finetuning (Visible overfitting)

- *Zbior testowy* : FruitRecognition
- *Skutecznosc modelu(TOP5)* : 0.8677
- *Ilosc danych testowych* : 6705



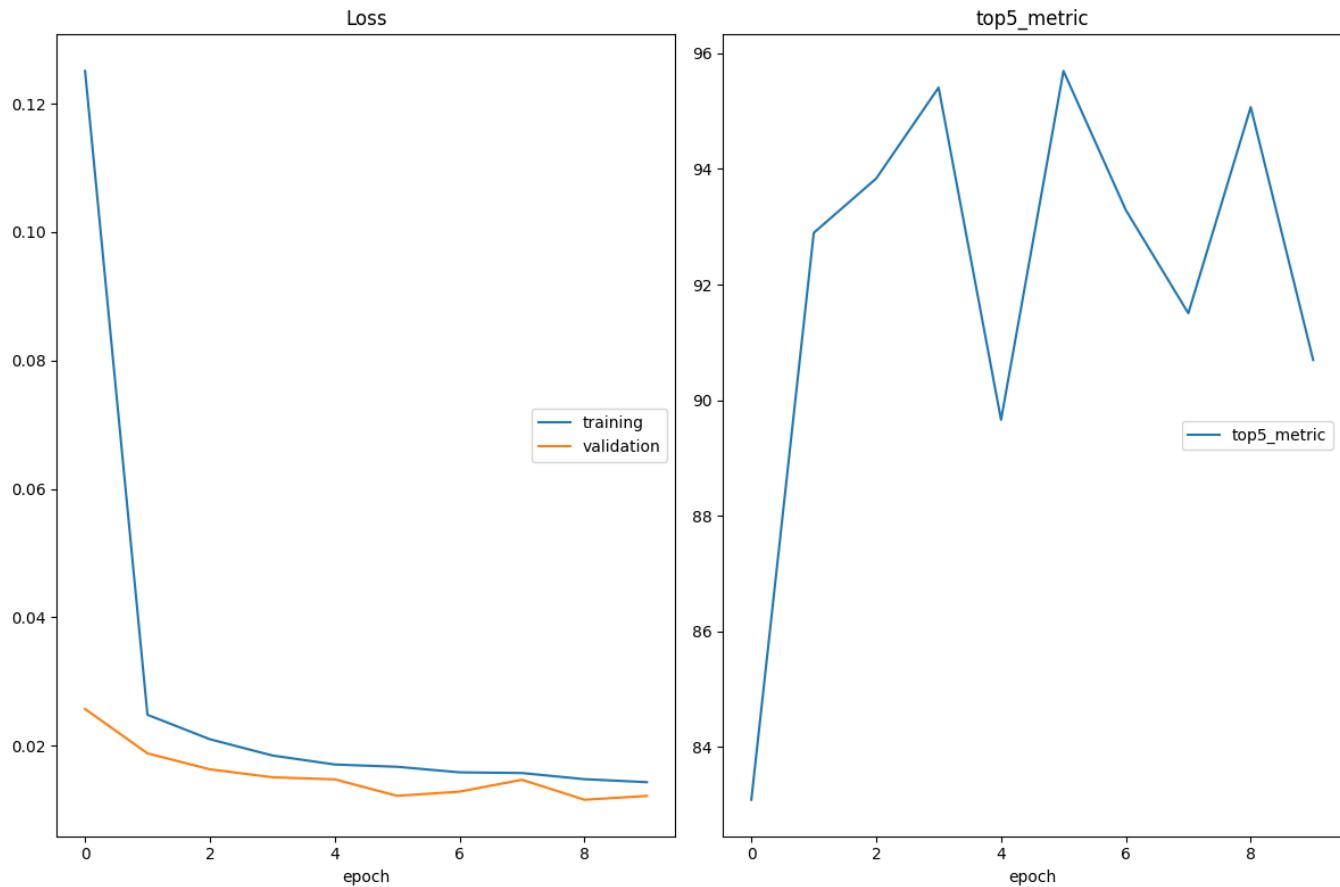
Rysunek 19: After finetuning (Visible overfitting)

Finetuning:

- **Warstwy przetrenowane :** [-4:]
- **Optimizer :** SGD 0.001 momentum 0.9
- **Skutecznosc modelu(TOP5) :** 0.8419
- **Ilosc danych testowych :** 6705

2.14. Test Dense512 Dense16 Contrastive

- *Metoda* : Contrastive loss
- *Wartsty* : Dense 512, Dense 16
- *Transfer learning* : Tak
- *Model bazowy* : VGG16
- *Training pairs* : 5000
- *Validation pairs* : 500
- *Batch size* : 16
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : auto



Rysunek 20: Before finetuning

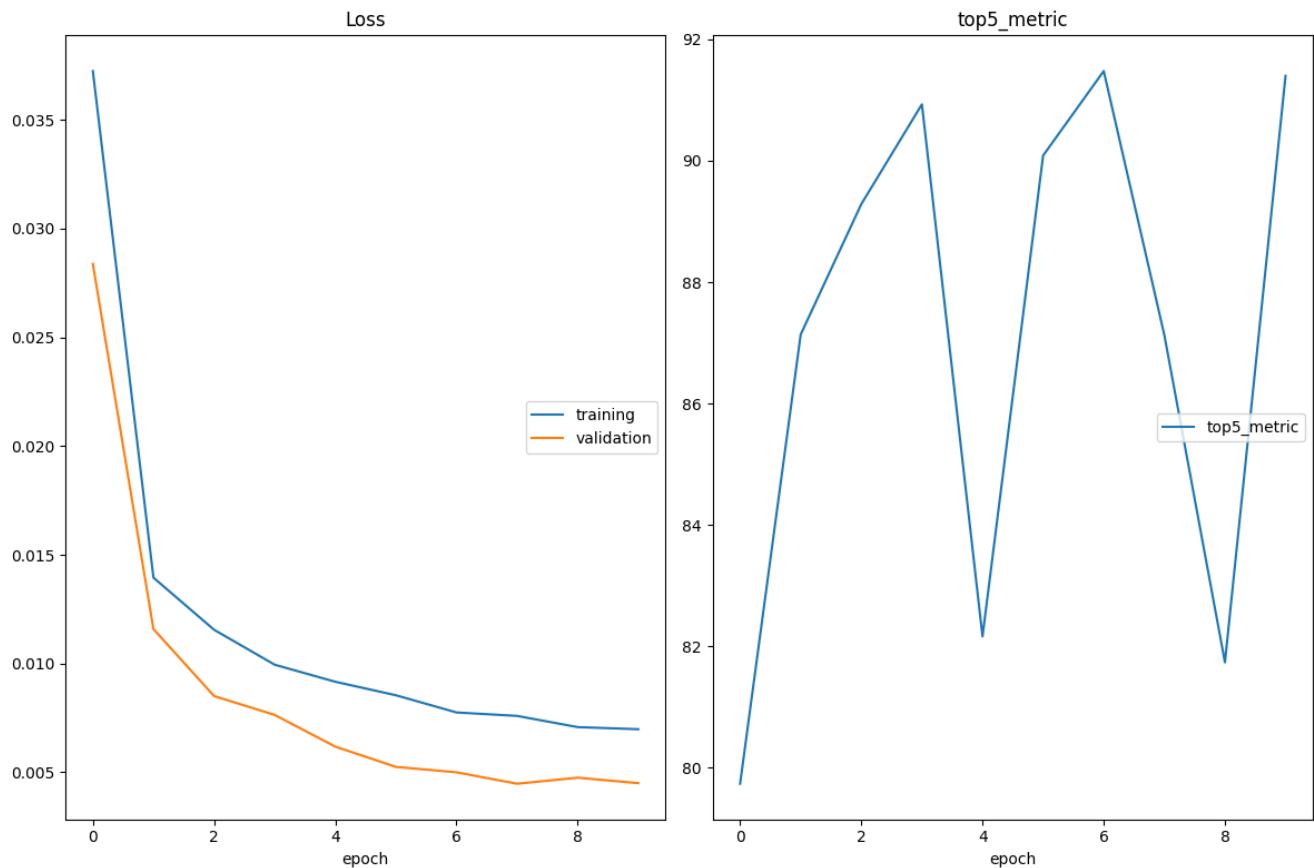
- *Ilosc danych testowych* : 250
Skutecznosc modelu : 90.69%
- *Ilosc danych testowych* : 500
Skutecznosc modelu : 91.06%
- *Ilosc danych testowych* : 6700
Skutecznosc modelu : 91.93%

Efficiency for Kiwi B is 95.8
 Efficiency for Orange is 97.42793791574289
 Efficiency for guava B is 99.37448559670781
 Efficiency for Apple B is 91.09090909090905
 Efficiency for Pomegranate is 83.91999999999994
 Efficiency for Apple C is 99.31999999999998
 Efficiency for Plum is 92.58720930232555
 Efficiency for Apple E is 94.84848484848486
 Efficiency for Carambola is 91.42765273311895
 Efficiency for kiwi A is 95.60486322188446
 Efficiency for Pear is 85.25942350332589
 Efficiency for Persimmon is 92.81935483870967
 Efficiency for guava A is 94.40707964601764
 Efficiency for Tomatoes is 74.50461538461536
 Efficiency for Apple A is 94.99029126213593
 Efficiency for Apple D is 94.11688311688312
 Efficiency for muskmelon is 91.87781350482308
 Efficiency for Peach is 76.46192893401009
 Efficiency for Banana is 95.42731277533048
 Efficiency for Apple F is 84.86842105263148
 Efficiency for Pitaya is 97.97866666666667
 Efficiency for Mango is 93.13001605136463
 Efficiency for Kiwi C is 97.34640522875813
 Efficiency of model in our metric: 91.93868498584546

Rysunek 21: Efektywnośc dla każdej z klas

2.15. Test Dense512 Dense16 Contrastive

- **Metoda** : Contrastive loss
- **Wartsty** : Dense 512, Dense 16
- **Transfer learning** : Tak
- **Model bazowy** : VGG16
- **Trainingpairs** : 31000
- **Validation pairs** : 6700
- **Barch size** : 16
- **Epoki** : 10
- **Imagesize** : 224x224
- **Margin** : 0.5
- **Liczba reprezentantow** : 5
- **Optimizer** : auto



Rysunek 22: Before finetuning. 31 tys trains, 6.7tys validation

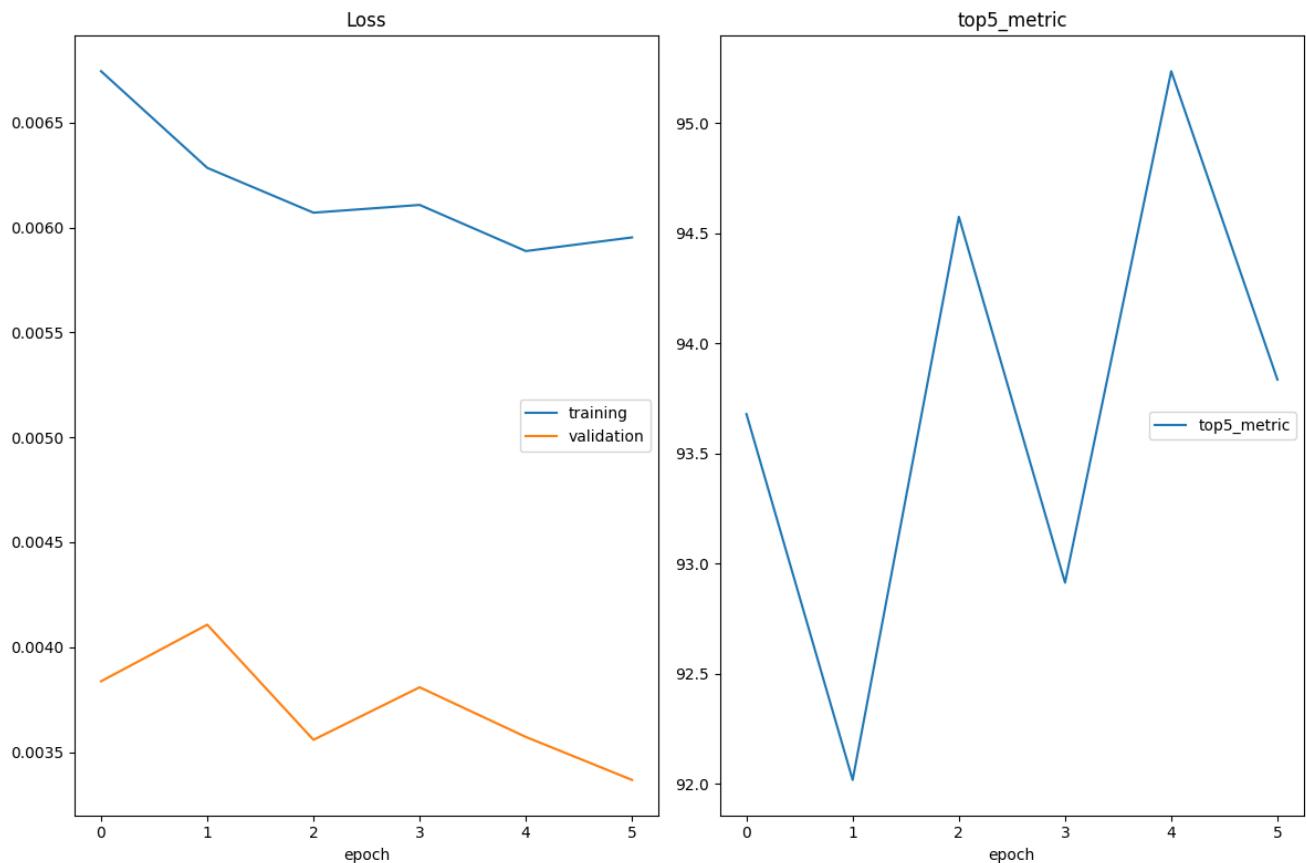
- *Ilosc danych testowych* : 250
Skutecznosc modelu : 91.40%
- *Ilosc danych testowych* : 500
Skutecznosc modelu : 89.32%
- *Ilosc danych testowych* : 6700
Skutecznosc modelu : 90.73%

Efficiency for Kiwi B is 84.47499999999998
 Efficiency for Orange is 95.30820399113088
 Efficiency for guava B is 96.88888888888889
 Efficiency for Apple B is 77.85454545454546
 Efficiency for Pomegranate is 83.81538461538457
 Efficiency for Apple C is 92.49333333333337
 Efficiency for Plum is 99.296511627907
 Efficiency for Apple E is 90.707070707070707
 Efficiency for Carambola is 91.4855305466238
 Efficiency for kiwi A is 92.55927051671725
 Efficiency for Pear is 94.25720620842556
 Efficiency for Persimmon is 87.45806451612897
 Efficiency for guava A is 94.60176991150435
 Efficiency for Tomatoes is 75.69230769230761
 Efficiency for Apple A is 93.02912621359218
 Efficiency for Apple D is 88.42857142857143
 Efficiency for muskmelon is 95.76205787781343
 Efficiency for Peach is 74.0406091370558
 Efficiency for Banana is 95.86784140969165
 Efficiency for Apple F is 91.15789473684207
 Efficiency for Pitaya is 99.328000000000002
 Efficiency for Mango is 94.69983948635665
 Efficiency for Kiwi C is 97.7908496732026
 Efficiency of model in our metric: 90.73903817274325

Rysunek 23: Efektywność dla każdej z klas

2.16. Test Dense512 Dense16 Contrastive FINE TUTNING

- *Metoda* : Contrastive loss
- *Wartsty* : Dense 512, Dense 16
- *Transfer learning* : Tak
- *Model bazowy* : VGG16
- *Trainingpairs* : 10000
- *Validation pairs* : 2700
- *Barch size* : 16
- *Epoki* : 6
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : SGD lr = 0.001



Rysunek 24: FINE TUTNIG 4 layers. 10 tys trains, 2.7tys validation

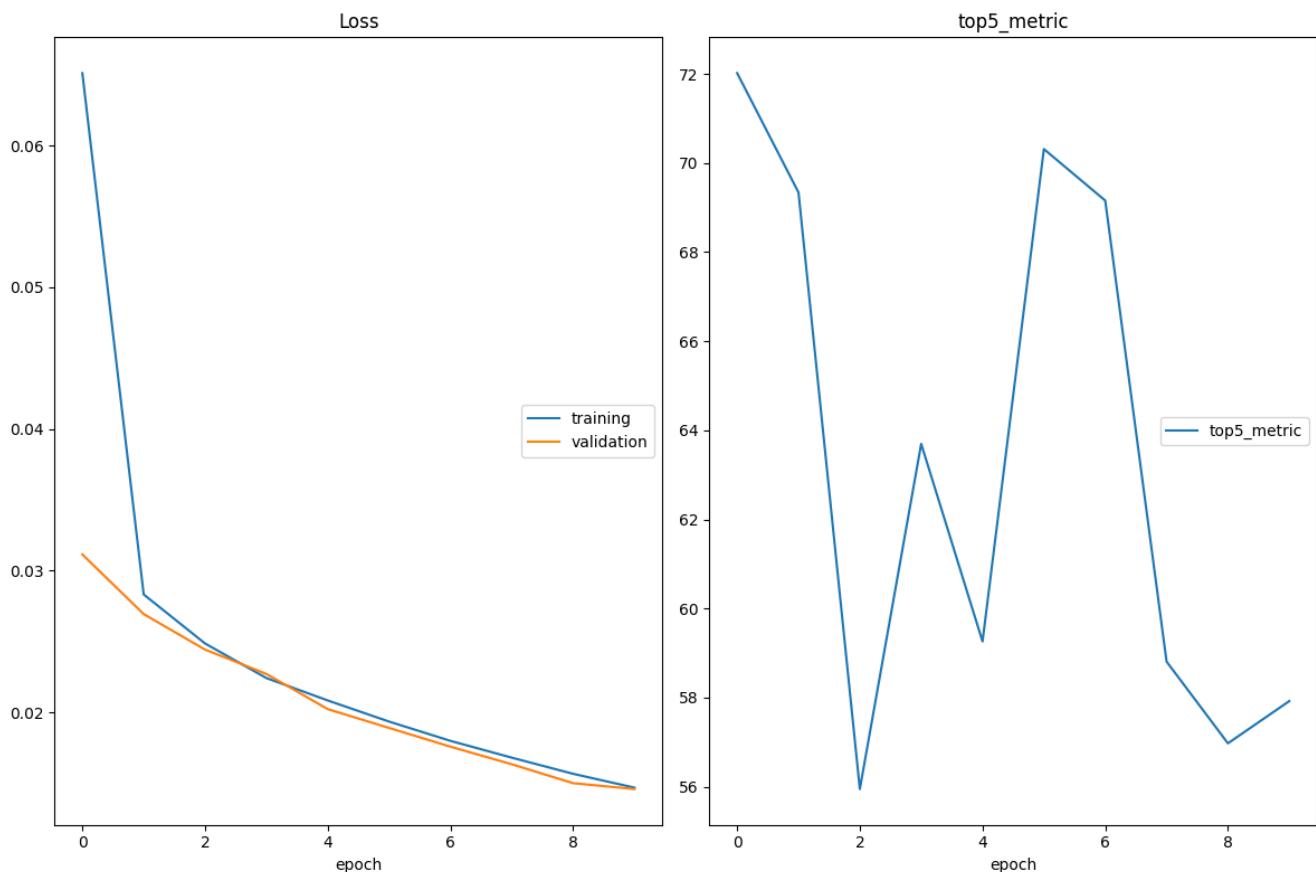
- *Ilosc danych testowych* : 250
Skutecznosc modelu : 93.83%
- *Ilosc danych testowych* : 500
Skutecznosc modelu : 94.69%
- *Ilosc danych testowych* : 6700 (wszystkie)
Skutecznosc modelu : 93.22%

Efficiency for Kiwi B is 91.87499999999999
Efficiency for Orange is 95.39689578713984
Efficiency for guava B is 98.97119341563784
Efficiency for Apple B is 95.14545454545456
Efficiency for Pomegranate is 86.22153846153842
Efficiency for Apple C is 93.46666666666664
Efficiency for Plum is 99.9593023255814
Efficiency for Apple E is 92.84848484848484
Efficiency for Carambola is 92.10289389067526
Efficiency for kiwi A is 97.3556231003039
Efficiency for Pear is 95.25942350332593
Efficiency for Persimmon is 93.19354838709674
Efficiency for guava A is 97.44247787610605
Efficiency for Tomatoes is 83.56307692307686
Efficiency for Apple A is 93.61165048543688
Efficiency for Apple D is 89.14285714285714
Efficiency for muskmelon is 96.05787781350476
Efficiency for Peach is 71.94923857868012
Efficiency for Banana is 95.89867841409692
Efficiency for Apple F is 93.64473684210522
Efficiency for Pitaya is 98.81066666666666
Efficiency for Mango is 92.91492776886052
Efficiency for Kiwi C is 99.42483660130718
Efficiency of model in our metric: 93.22856739324365

Rysunek 25: Efektywność dla każdej z klas

2.17. Dense256 Dense64 Contrastive - Training on VegFru (*1)

- **Metoda** : Contrastive loss
- **Wartsty** : Dense 256, Dense 64
- **Transfer learning** : Tak
- **Model bazowy** : Xception (Vegetables)
- **Trainingpairs** : 72000
- **Validation pairs** : 9000
- **Barch size** : 128
- **Epoki** : 10
- **Imagesize** : 299x299
- **Margin** : 0.5
- **Liczba reprezentantow** : 5
- **Optimizer** : auto



Rysunek 26: Before finetuning. 72 tys trains, 9tys validation

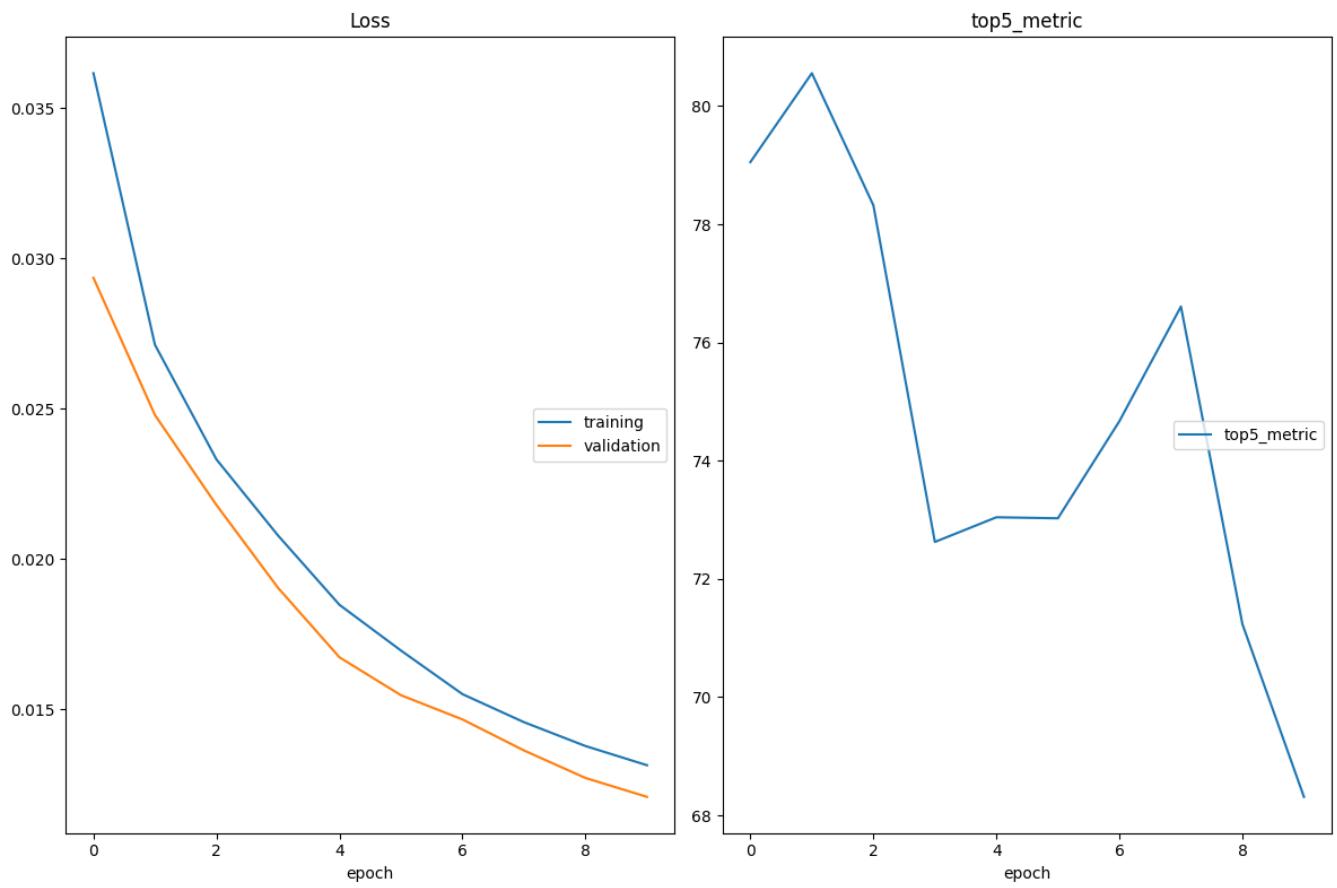
- *Ilosc danych testowych* : 250
Skutecznosc modelu : 58%
- *Ilosc danych testowych* : 9000
Skutecznosc modelu : 59%

```
Efficiency for Kiwi B is 47.587500000000001
Efficiency for Orange is 80.66962305986694
Efficiency for guava B is 62.13168724279829
Efficiency for Apple B is 71.67272727272722
Efficiency for Pomegranate is 58.82461538461536
Efficiency for Apple C is 37.239999999999995
Efficiency for Plum is 77.84883720930225
Efficiency for Apple E is 58.30303030303031
Efficiency for Carambola is 76.52733118971045
Efficiency for kiwi A is 58.49848024316104
Efficiency for Pear is 61.85809312638574
Efficiency for Persimmon is 50.096774193548356
Efficiency for guava A is 47.5663716814159
Efficiency for Tomatoes is 76.70153846153845
Efficiency for Apple A is 55.650485436893184
Efficiency for Apple D is 43.42857142857143
Efficiency for muskmelon is 62.03858520900312
Efficiency for Peach is 59.74619289340094
Efficiency for Banana is 80.77973568281934
Efficiency for Apple F is 17.868421052631582
Efficiency for Pitaya is 64.27199999999998
Efficiency for Mango is 61.92616372391654
Efficiency for Kiwi C is 43.973856209150306
Efficiency of model in our metric: 58.92220091323854
```

Rysunek 27: Efektywność dla każdej z klas

2.18. Dense256 Dense64 Contrastive - Training on VegFru (*2)

- *Metoda* : Contrastive loss
- *Wartsty* : Dense 256, Dense 64
- *Transfer learning* : Tak
- *Model bazowy* : Xception (Fruits)
- *Training pairs* : 50000
- *Validation pairs* : 6500
- *Batch size* : 128
- *Epoki* : 10
- *Imagesize* : 299x299
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : auto



Rysunek 28: Before finetuning. 50 tys trains, 6.5tys validation

- *Ilosc danych testowych* : 250
Skutecznosc modelu : 68.5%
- *Ilosc danych testowych* : 9000
Skutecznosc modelu : 69.6%

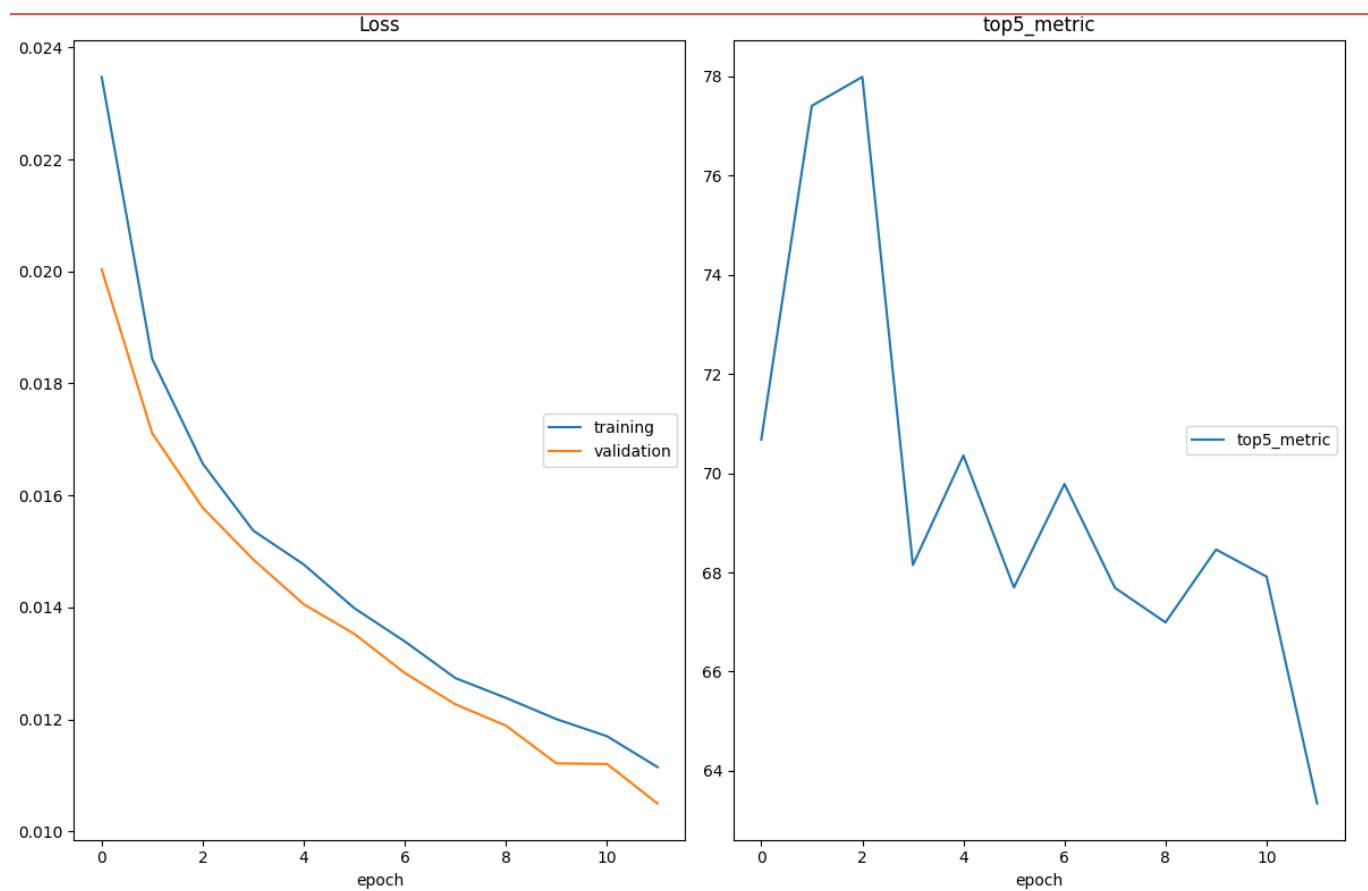
Przed treningiem załadowane wagi z poprzedniego treningu (*1)

```
Efficiency for Kiwi B is 38.449999999999999
Efficiency for Orange is 97.179600886918
Efficiency for guava B is 67.54732510288058
Efficiency for Apple B is 99.70909090909092
Efficiency for Pomegranate is 57.63692307692298
Efficiency for Apple C is 79.94666666666667
Efficiency for Plum is 72.29651162790691
Efficiency for Apple E is 59.33333333333333
Efficiency for Carambola is 98.68167202572349
Efficiency for kiwi A is 51.90273556231002
Efficiency for Pear is 75.99113082039915
Efficiency for Persimmon is 86.30322580645148
Efficiency for guava A is 53.66371681415924
Efficiency for Tomatoes is 85.68615384615372
Efficiency for Apple A is 28.73786407766992
Efficiency for Apple D is 72.55844155844157
Efficiency for muskmelon is 34.63022508038583
Efficiency for Peach is 77.98477157360402
Efficiency for Banana is 98.57268722466962
Efficiency for Apple F is 81.36842105263156
Efficiency for Pitaya is 97.834666666666668
Efficiency for Mango is 54.83146067415727
Efficiency for Kiwi C is 36.444444444444446
Efficiency of model in our metric: 69.88222038398206
```

Rysunek 29: Efektywność dla każdej z klas

2.19. Dense256 Dense64 Contrastive - Training on VegFru (*3)

- *Metoda* : Contrastive loss
- *Wartsty* : Dense 256, Dense 64
- *Transfer learning* : Tak
- *Model bazowy* : Xception (Vegetables + Fruits)
- *Trainingpairs* : 125000
- *Validation pairs* : 16000
- *Barch size* : 128
- *Epoki* : 11
- *Imagesize* : 299x299
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : auto



Rysunek 30: Before finetuning. 125 tys trains, 16tys validation

- **Ilosc danych testowych :** 250
Skutecznosc modelu : 58%
- **Ilosc danych testowych :** 9000
Skutecznosc modelu : 59%

Przed treningiem załadowane wagi z poprzedniego treningu (*2)

Wnioski z przeprowadzonych treningów na zbiorze VegFru:

- Chociaż zbiór VegFru jest bardzo dużym zbiorem owoców i warzyw to wyniki są słabe, nawet przy bardzo długim treningu
- Model uczy się zbioru VegFru (dlatego Loss na wykresie maleje)
- Zbiór VegFru jest bardzo różnorodny. Owoce tego samego gatunku przedstawione są w najróżniejszej formie (na drzewie, w koszyku, w reklamówce, o bardzo różnych skrajnych dla tego gatunku kolorach, formach podania, przekrojeniach, zgniecenia).
- Ten zbiór jest ciężki do nauczenia z racji swojej różnorodności. Natomiast wystarczająco dobrze wytrenowany model będzie osiągał dobre wyniki na tym zbiorze VegFru.
- Testy na FruitRecognition: Wytrenowany model (*2), dobrze rozpoznaje owoce które się wyróżniają od pozostałych - Banany 98.5% , Pitaya 97.8%. Są to na tyle rozróżnialne owoce, że model nie ma z nimi najmniejszego problemu.
- Sytuacja jest dużo słabsza, gdy trzeba rozróżnić Kiwi A, Kiwi B, Kiwi C. Są to bardzo podobne owoce, które wymagają, aby model bardzo dokładnie zapoznał się z ich strukturą, kształtem i podobnymi. Tak aby mógł wyłapywać te bardzo małe różnice.
- Model trenowany na VegFru, najprawdopodobniej dobrze nauczył się ignorować tło
- Prawdopodobnie problem w słabych wynikach testowych na zbiorze FruitRecognition leży właśnie w tej różnorodności poszczególnych gatunków w zbiorze uczącym VegFru.
- Potrzebujemy rozróżnić owoce całe, które jeśli są tego samego gatunku, zawsze są podobne do siebie. A model uczył się tego że ten sam gatunek owocu może wyglądać bardzo różnie. Dochodząc do sytuacji w których sporo zdjęć różnych gatunków owoców było dużo bardziej podobnych niż owoce z tego samego gatunku.
- Bazą do nauki modelu w naszym przypadku powinno być względnego podobieństwa (model nie musi sobie dodatkowo radzić z problemem różnorodności tła). Oraz jeden gatunek owocu powinien być w tej samej formie - (całe owoce, położone luźno). W sklepie owoce w obrębie jednego gatunku są zawsze podobne do siebie "mają jedną naturę". W ten sposób model który będzie uczył się na takim zbiorze, jego "wiedza" czyli konstruowana n-wymiarowa przestrzeń będzie tak "układała" poszczególne gatunki, aby rozróżnić je między sobą po ich konkretnych cechach (nawet niewiele różniących się między innymi gatunkami). Dzięki temu nowe wprowadzone gatunki do bazy danych, także będą tak rozróżnialne.
- W taki sposób model będzie rozróżniał nowy wprowadzony produkt "a" od innego wprowadzonego produktu "A", po tym że (jedyna różnica) ten drugi jest np lekko bardziej podłużny.

2.20. Propozycje stworzenia dobrego zbioru dla naszego problemu (na podstawie powyższych wniosków) :

- Zbiór powinien zawierać owoce których gatunki są ”podobnej natury”. Daje to możliwość modelowi wgłębiania się w mniej widoczne cechy rozróżniające każdy z produktów.
- Im zbiory uczące będą bardziej podobne do tych sklepowych warunków tym bardziej model będzie uczył się faktycznych różnic które będą dla niego przydatne. Będzie lepiej wgłębiał się w szczegóły - które przecież w naszym przypadku są bardzo ważne (różnice pomiędzy Kiwi A, Kiwi B, Kiwi C) są bardzo małe. - Tak samo będzie w przypadku gdy wprowadzimy nowe Kiwi D. Model musi umieć wchodzić w szczegóły produktów.
- Zbiór powinien zawierać takie zdjęcia które, nie będą zmuszały modelu do np (uczenia się wydzielenia owoców od tła) - pownieważ jest to coś czego musi się nauczyć, i jego potencjał na naukę innych ważnych w naszym przypadku rzeczy maleje.

3. Ostatni test

3.1. Dense128 Contrastive - Training on VegFru

- *Metoda* : Contrastive loss
- *Wartsty* : Dense 128
- *Transfer learning* : Tak
- *Model bazowy* : InterceptionResNetV2(Fruits)
- *Trainingpairs* : 10000
- *Validation pairs* : 2000
- *Barch size* : 128
- *Epoki* : 10
- *Imagesize* : 224x224
- *Margin* : 0.5
- *Liczba reprezentantow* : 5
- *Optimizer* : auto

Layer (type)	Output Shape	Param #	Connected to
input_layer_2 (InputLayer)	(None, 224, 224, 3)	0	-
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0	-
functional_1 (Functional)	(None, 128)	59,252,064	input_layer_2[0]... input_layer_3[0]...
lambda_1 (Lambda)	(None, 1)	0	functional_1[0]... functional_1[1]...

Total params: 59,252,064 (226.03 MB)

Trainable params: 4,915,328 (18.75 MB)

Non-trainable params: 54,336,736 (207.28 MB)

Rysunek 31: Before finetuning. 10 tys trains, 2tys validation

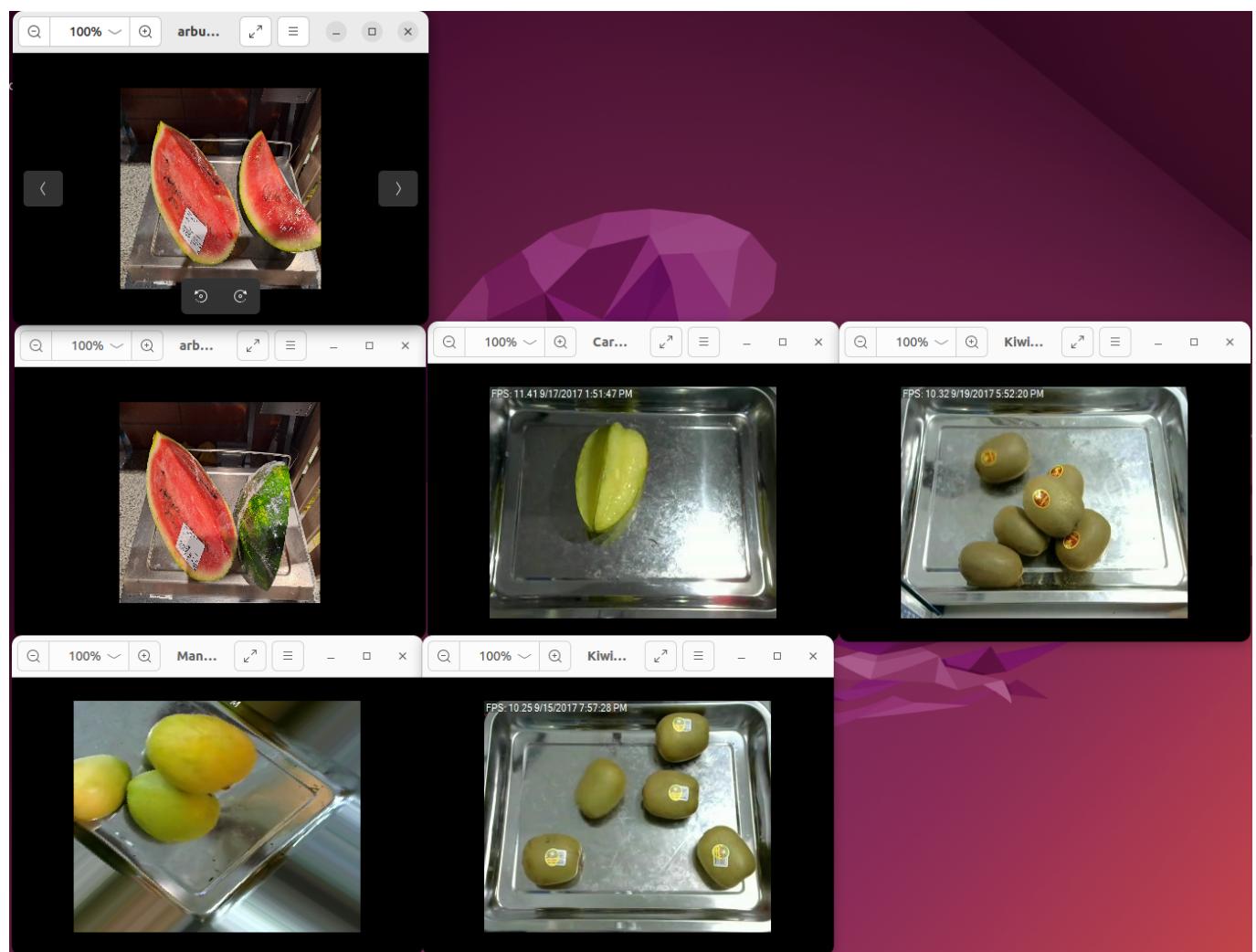
Efficiency of model in our metric: 80.06521739130432

Rysunek 32: Efficiency of our model

We make few photos in shop, of 2 new classes of fruits to our representants database. Then we put to trained model photo of this fruit and see the results.

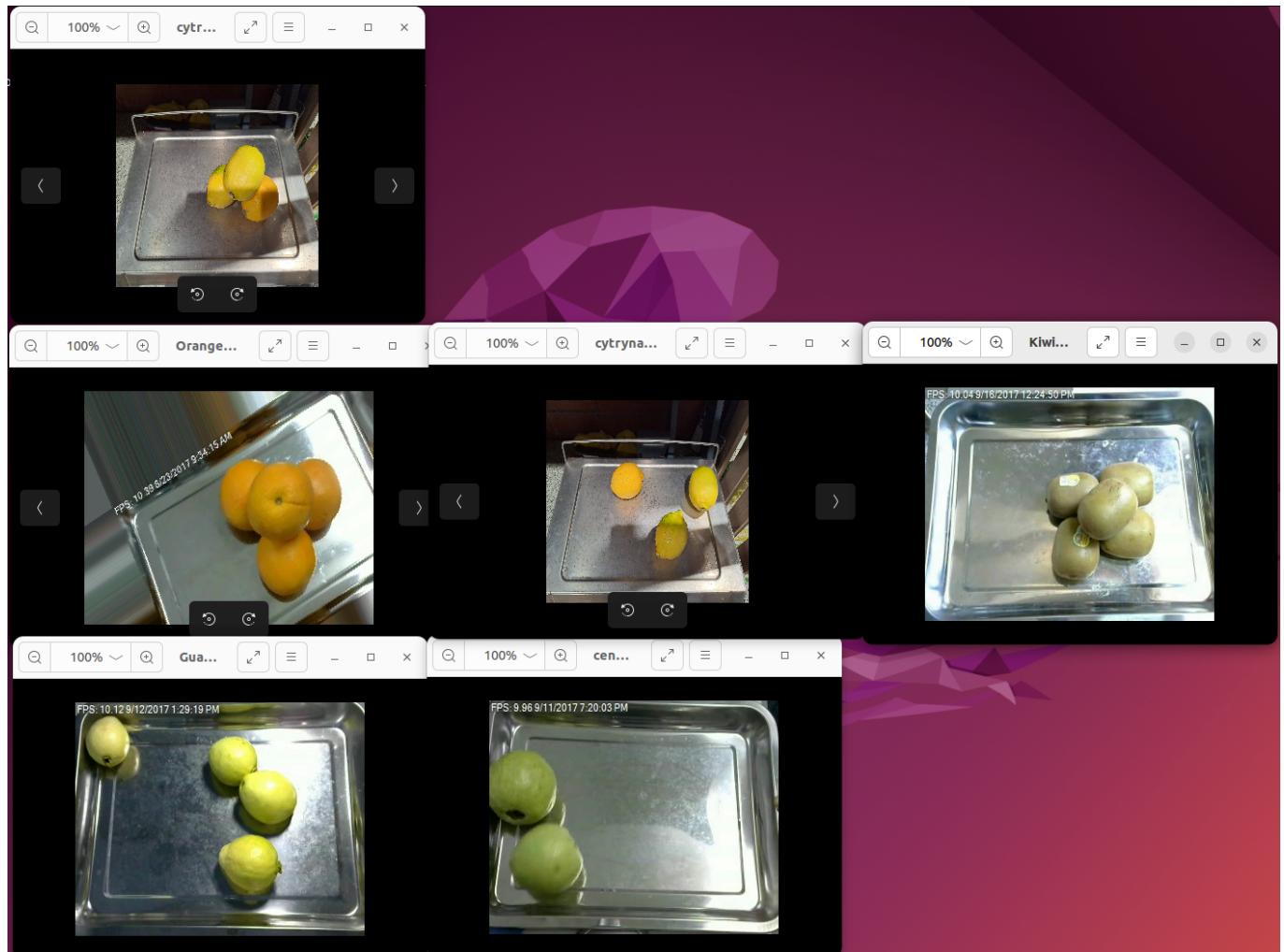
```
[0.05023064 0.12891738 0.13562357 0.13658249 0.15698954]  
['Arbuz_copyright_small', 'Carambola', 'Kiwi B', 'Mango', 'kiwi A']  
Arbuz_copyright_small
```

Rysunek 33: Put arbuz to model, get embedding and find the most similar products in database of representants



Rysunek 34: Model znalazł top5 reprezentantów dla arbuzu

Z koleją gdy wrzuciliśmy cytrynę do modelu, aby znalazł nam najbliższych reprezentantów, uzyskaliśmy takie wyniki:



Rysunek 35: Model znalazł top5 reprezentantów dla cytryny. Tylko pomarańcze uzyskały lepszy wynik