



Projekt GOLEM

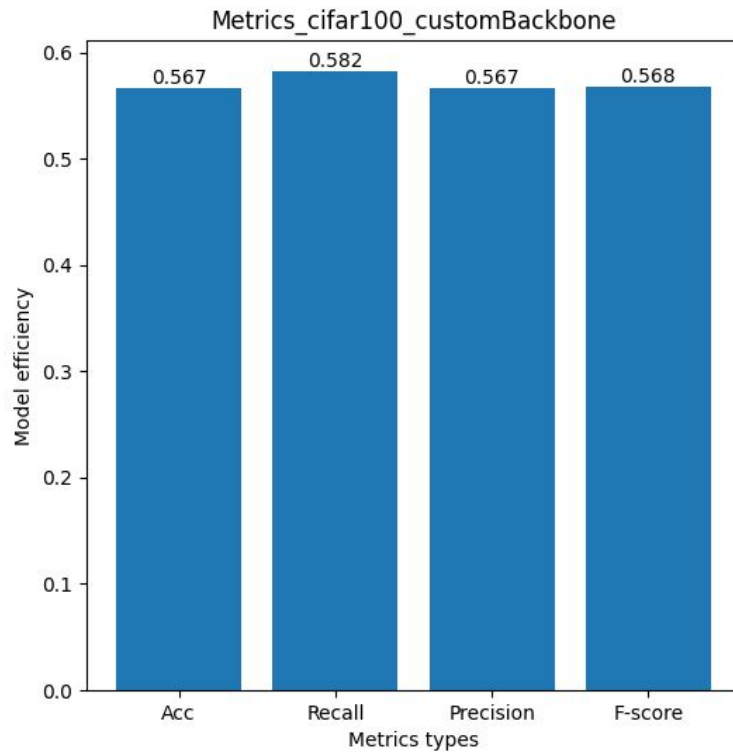
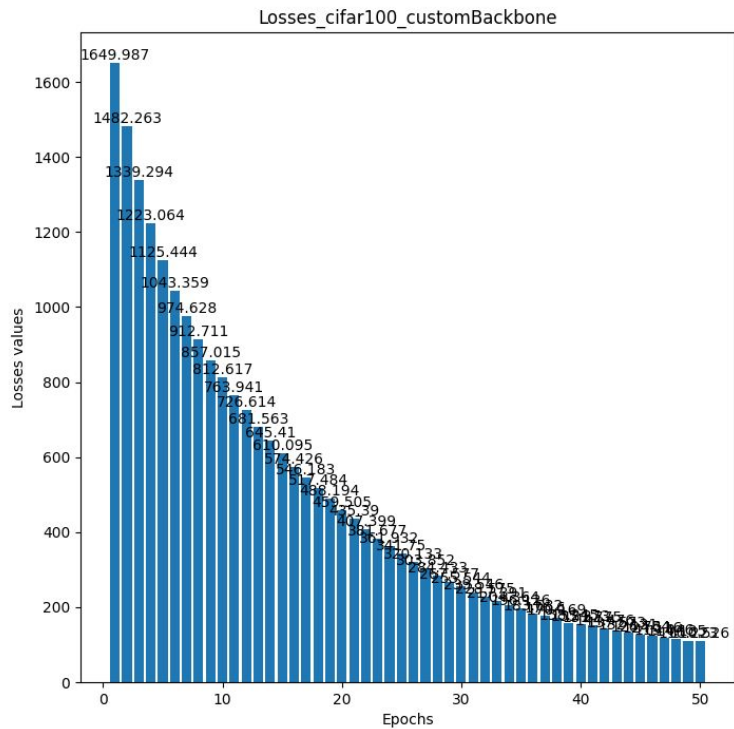
Sieci CNN - Klasyfikacja obrazów

Jakub Kwaśniak/Mieczkowski



Trening customowych sieci CNN

- Trening na natywnym rozmiarze CIFAR-100, 32x32
- Próba treningu na 224x224 zakończona niepowodzeniem - zbyt długi czas nauki
- Krótki czas treningu dla 32x32
- Strata wygląda na funkcję homograficzną
- Dokładność w okolicach 0.6
- Do klasyfikacji używany bardzo prosty klasyfikator = warstwa fully connected



Struktura Sieci Golem Backbone 3


- Najlepszy uzyskany przez nas model
- Architektura zbliżona do ResNet



Dobór parametrów

- Specjalna funkcja do doboru najlepszych parametrów dla danego modelu
- Dla modelu GB3:
 - Funkcja straty: CrossEntropyLoss
 - Optymalizator: Adam
 - Learning Rate rzędu 0.001
- Dla ResNet18:
 - Funkcja straty i optymalizator te same
 - Używając Linear Probing LR najlepszy 0.001
 - Fine Tuning 0.0001
 - Logiczne, bo przy Fine-Tuningu nie chcemy przeuczyć pretrenowanego modelu

```
Test for optim <class 'torch.optim.adam.Adam'> and loss func <class 'torch.nn.modules.loss.CrossEntropyLoss'>:
Learning rate: 0.0001
1/3: loss=2721.3335809111595
2/3: loss=2280.4084282517433
3/3: loss=2175.8393872976303
Learning rate: 0.001
1/3: loss=2711.481388449669
2/3: loss=2189.593093574047
3/3: loss=2033.4878758788109
Learning rate: 0.01
1/3: loss=3584.3027963638306
2/3: loss=3987.6956086158752
Learning rate: 0.1
1/3: loss=5893777.345420003
2/3: loss=68060.11453294754
3/3: loss=284770.0880883932
Test for optim <class 'torch.optim.sgd.SGD'> and loss func <class 'torch.nn.modules.loss.CrossEntropyLoss'>:
Learning rate: 0.0001
1/3: loss=8640.188005566597
2/3: loss=3869.8778800964355
3/3: loss=3490.1635591983795
Learning rate: 0.001
1/3: loss=nan
Learning rate: 0.01
1/3: loss=nan
...
Optim lr: 0.001
Loss: 2033.4878758788109
Best optim: <class 'torch.optim.adam.Adam'>
Best loss func: <class 'torch.nn.modules.loss.CrossEntropyLoss'>
```



Własny trening (wag) sieci neuronowych na zbiorze CIFAR-100

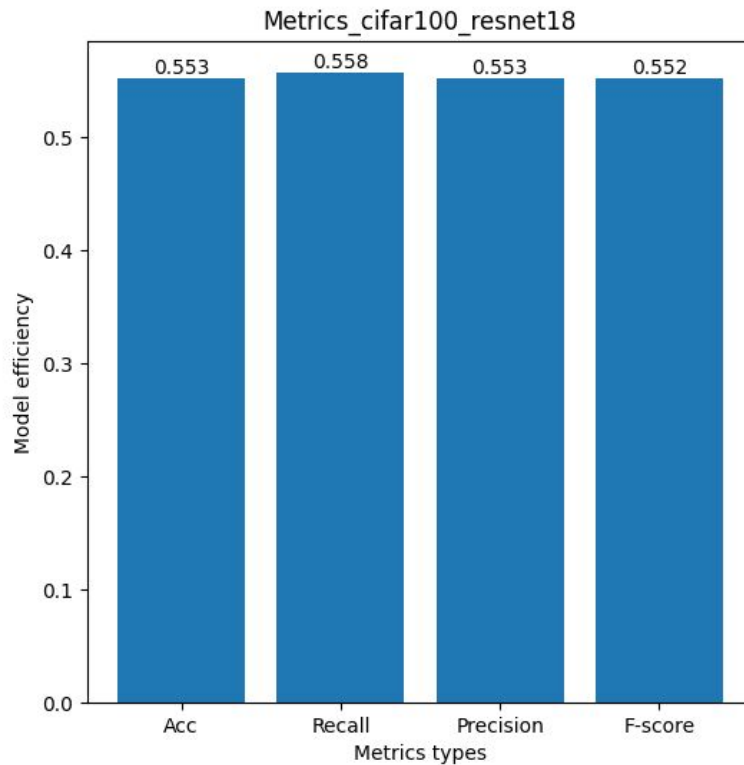
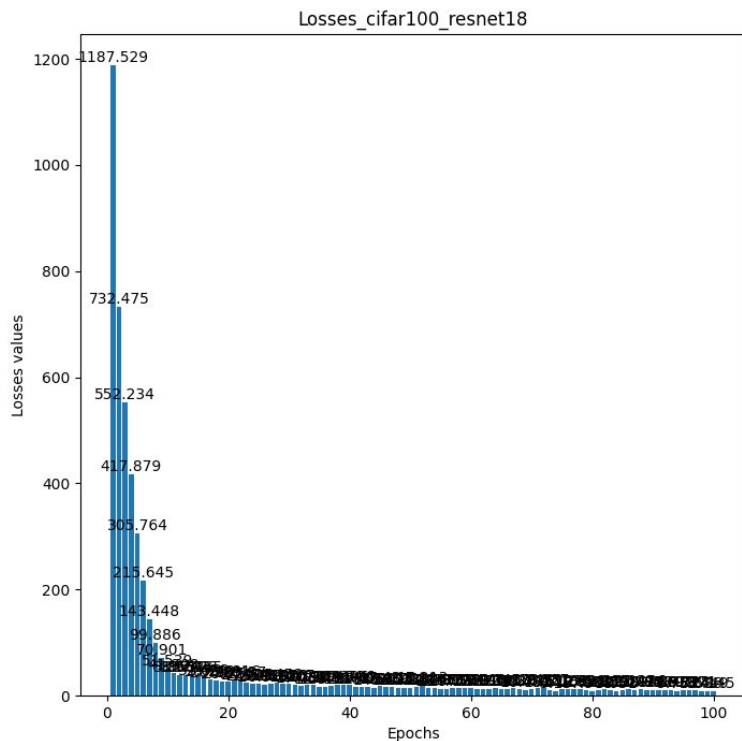
- Domyślny rozmiar danych w dataset CIFAR-100: 60 000 obrazów 32px x 32px (100 klas)
- Mały batchsize = 32 daje bardzo słabe wyniki treningu, na poziomie 0.2-0.3 dokładności, większy trochę lepsze lecz wciąż niezadowalające
- Najlepszy customowy model radzi sobie równie dobrze/lepiej niż modele Resnet/ViTBase



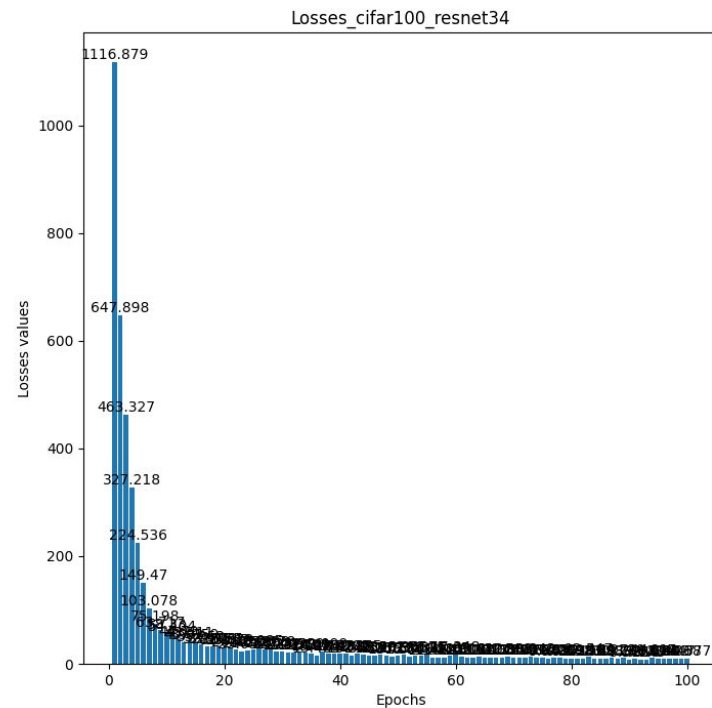
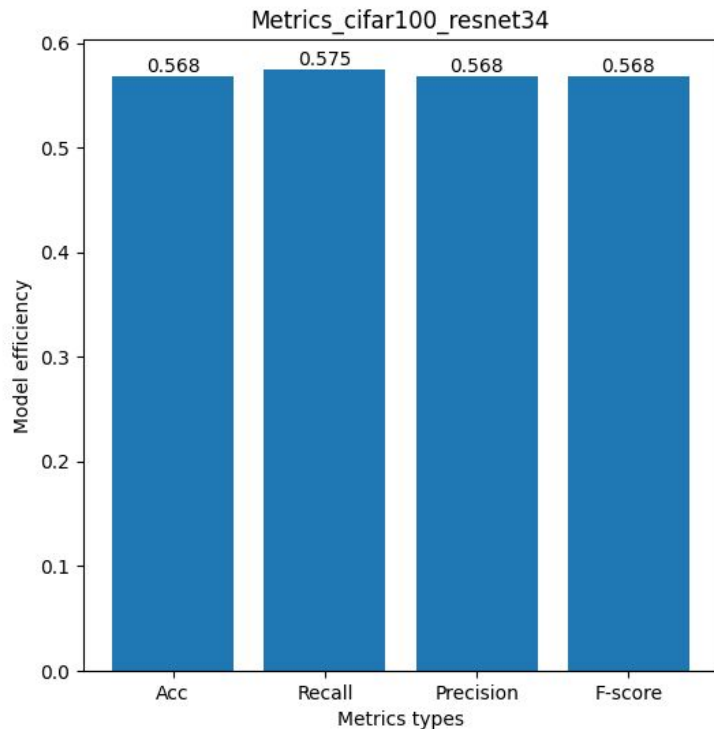
Testy pretrenowanego ResNet-18 na zbiorze CIFAR-100:

- Trenowanie klasyfikatora ResNet metodą Linear Probing (z wagami pretrenowanymi) na dataset o standardowych wymiarach zdjęć 32x32px daje dokładność rzędu 0.2-0.3 - wynika to stąd, że modele ResNet są na danych o rozm. 224x224px
- Fine-Tuning daje już sensowne wyniki, w okolicach 0.5 - do trenowanie wag pozwala na ich lepsze przystosowanie do danych o innych parametrach niż pierwotne
- Skalowanie danych do rozmiaru 224x224 wyniki poprawiają się
- ResNet-18 Linear Probing dochodzi do 0.58 dokładności
- Wersja Fine-Tuning osiąga wyniki w okolicach 0.8 dokładności
- Przy metodzie FT model względnie szybko dostraja wagi, przy 8 epoce przerwaliśmy uczenie gdyż rezultat był satysfakcjonujący a dalej następowało przeuczenie
- Przy metodzie LP tego problemu nie zauważyliśmy

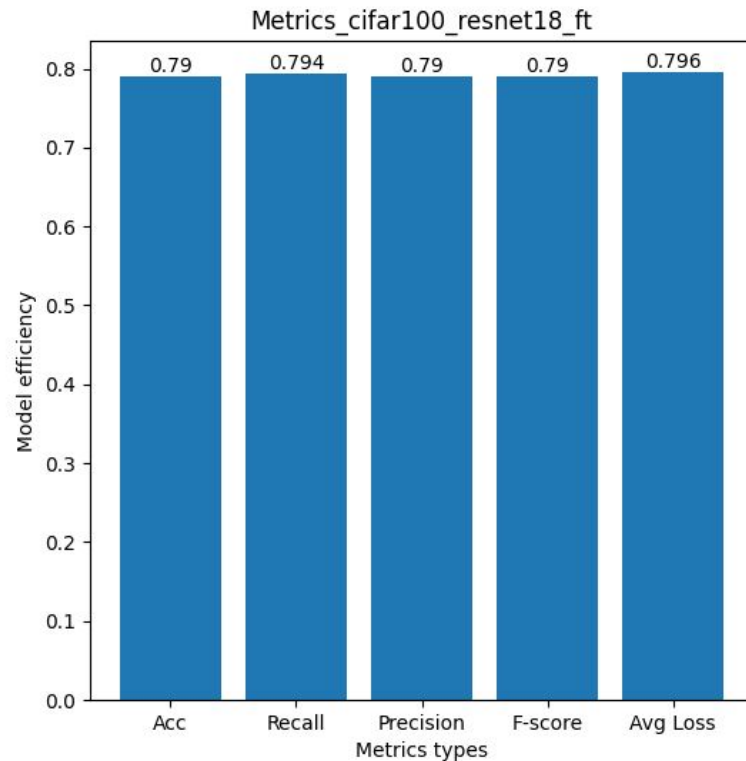
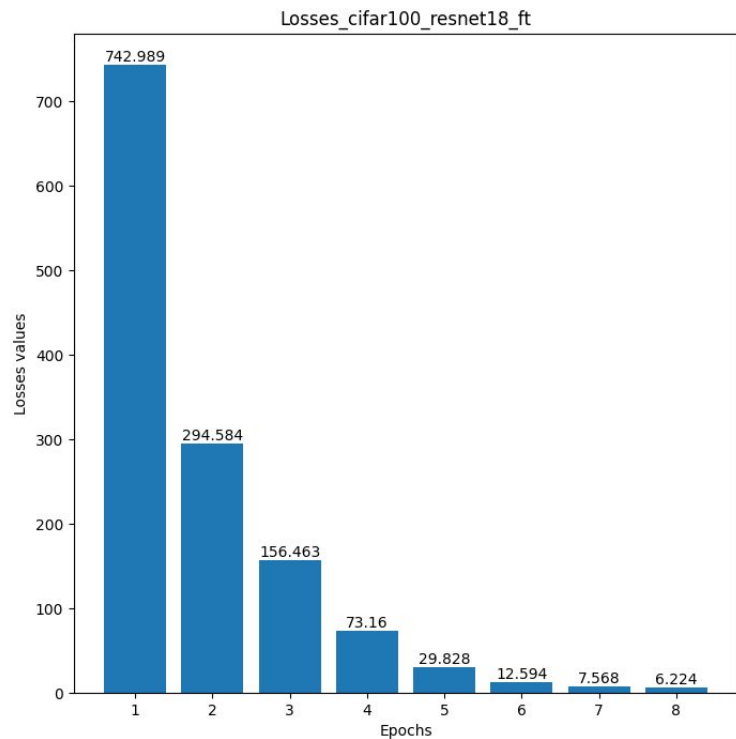
Fine tuning na datasetcie o rozm. danych 32x32px



Fine tuning na datasetcie o rozm. danych 32x32px



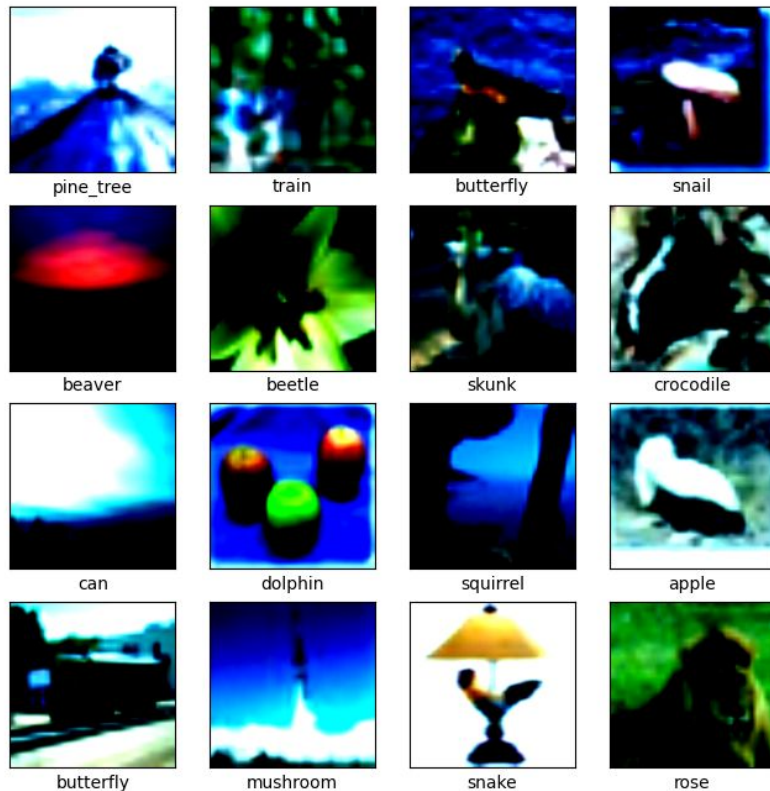
Fine tuning na datasecie o rozm. danych 224x224px



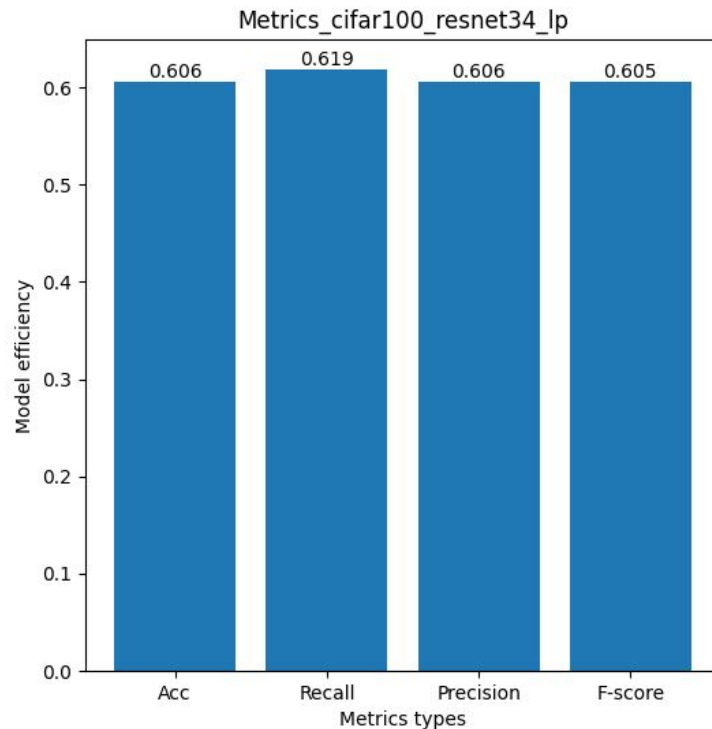
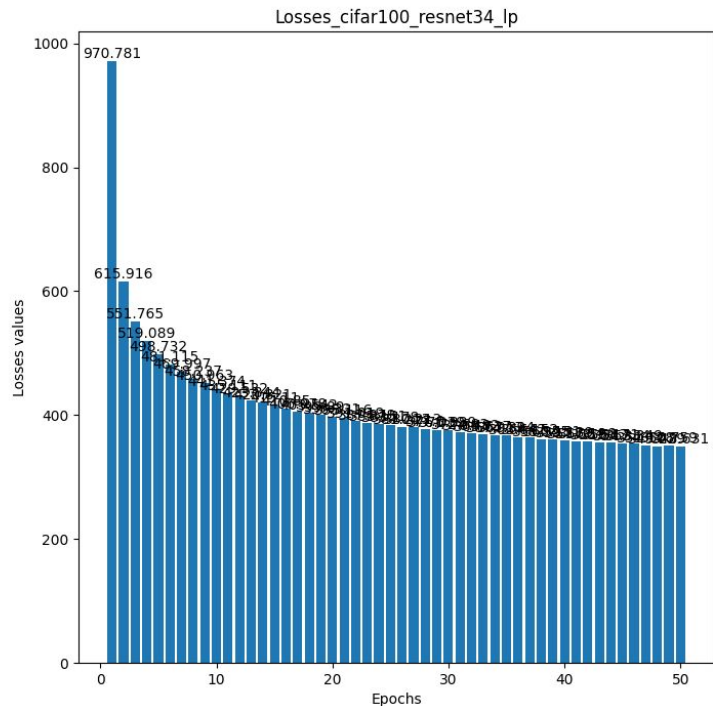
Przykładowy wynik klasyfikacji

- FT ResNet-18

-dataset 224x224px

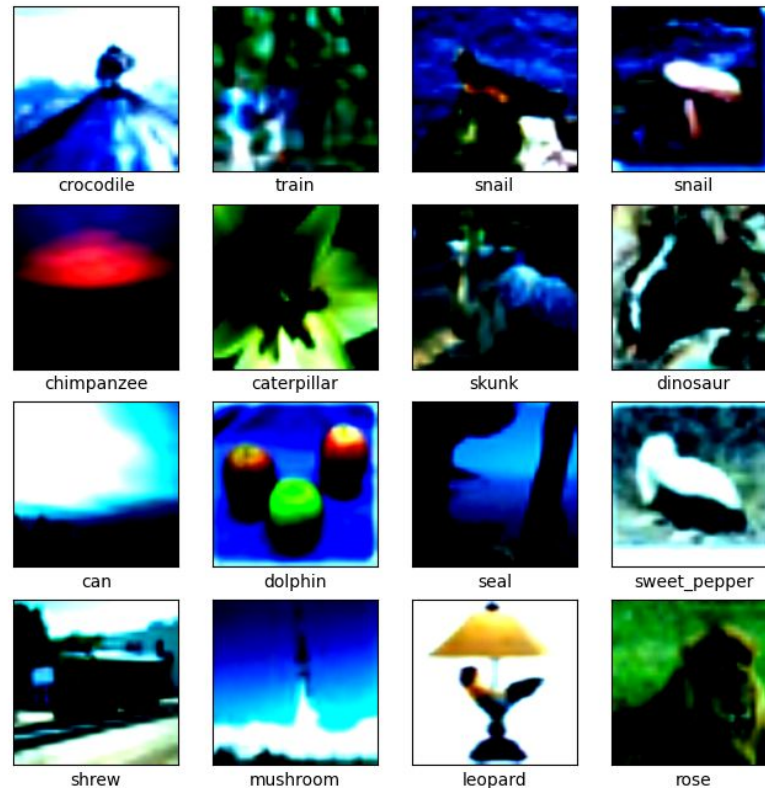


Linear Probing na datasecie o rozm. danych 224x224px



Przykładowy wynik klasyfikacji

- LP ResNet-34
- dataset 224x224px





Dziękujemy za uwagę