

# ANALIZA I BADANIE DZIAŁANIA WYBRANYCH SIECI NEURONOWYCH W ZADANIU KLASYFIKACJI OBRAZÓW NA ZBIORZE IMAGENET

AUTOR: MAREK SIGMUND

PROMOTOR PRACY: DR HAB. INŻ. PROF. PCZ KRYSTIAN ŁAPA

# SPIS TREŚCI

## 1. Wstęp

- Wprowadzenie do tematu klasyfikacji, motywacja oraz cel i zakres badań.

## 2. Zbiór danych

- Opis zbiorów ImageNet i Tiny ImageNet oraz wstępne przetwarzanie danych.

## 3. Technologie i metodologia eksperymentów

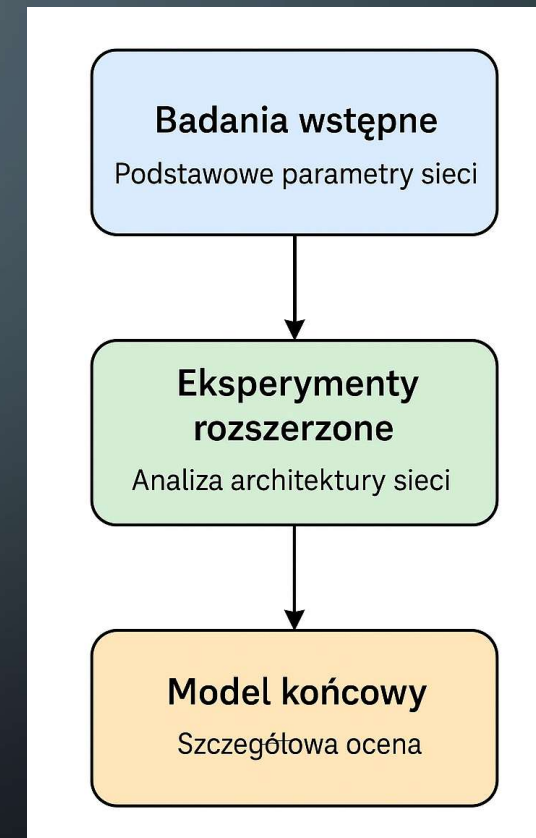
- Omówienie środowiska obliczeniowego oraz struktury eksperymentów.

## 4. Konwolucyjne sieci neuronowe (CNN)

- Teoretyczne podstawy działania CNN, przegląd architektur i zastosowań.

## 5. CZĘŚĆ EKSPERYMENTALNA – ANALIZA I PORÓWNIANIE ARCHITEKTUR SIECI CNN

- **Etap I** – Eksperymenty wstępne: struktura sieci i zachowanie modeli
  - analiza wpływu głębokości sieci, warstw FC, aktywacji i regularyzacji
- **Etap II** – Eksperymenty zaawansowane: zachowanie modeli podczas uczenia
  - badanie wpływu parametrów treningowych (LR, batch size, optymalizator) na trzech modelach o różnych profilach
- **Etap III** – Budowa finalnego modelu
  - projekt i trening modelu oparty na wnioskach z poprzednich etapów oraz literaturze



## 6. ANALIZA I PORÓWNIANIE GOTOWYCH ROZWIĄZAŃ

### DenseNet

- DenseNet121  
DenseNet161
- DenseNet169

### VGG

- VGG11      VGG13
- VGG16      VGG19

### Inne

- AlexNet
- GoogLeNet      ShuffleNet
- InceptionV3      SqueezeNet

### ResNet

- ResNet18      ResNet34
- ResNet50
- ResNet101      ResNet152

### MobileNet

- MobileNetV2 (Small i Large)
- MobileNetV3 (Small i Large)

### EfficientNet

- B0      B1
- B2      B3

# ZAKOŃCZENIE PRACY

- Praktyczne wykorzystanie najlepszych modeli
  - Opcjonalne testy działania najlepszego modelu własnego oraz wybranego modelu predefiniowanego w warunkach rzeczywistych (np. analiza obrazów z kamery).ego modelu i najlepszego z predefiniowanych
- Podsumowanie pracy
  - Omówienie rezultatów, najważniejszych wniosków oraz dalszych możliwych kierunków rozwoju.

# REZULTATY WŁASNYCH EKSPERYMENTÓW

W pierwszym etapie badań analizowano wpływ wybranych parametrów architektury sieci konwolucyjnych na stabilność procesu uczenia i jakość klasyfikacji. Zbadano:



- liczbę warstw konwolucyjnych i w pełni połączonych



- przypadki przeuczenia i niedouczenia



- wybór funkcji aktywacji



- oraz zastosowanie różnych technik regularyzacji

# LICZBA WARSTW KONWOLUCYJNYCH

- **ShallowCNN (2 warstwy)** – bardzo szybkie dopasowanie do treningu (**87% accuracy**), ale całkowite przeuczenie (**≈0% validacja**).

- **DeepCNN (12)** i **VeryDeepCNN (27)** – brak sensownej konwergencji, bardzo niska skuteczność, długi czas treningu.

Model	Liczba warstw konwolucyjnych	Train Accuracy	Validation Accuracy	Średni czas epoki	Konwergencja
ShallowCNN	2	87%	≈0%	~33.7 s	Tak (przeuczenie)
DeepCNN	12	0.5%	0%	~53.9 s	Nie
VeryDeepCNN	27	0.48%	0%	~86.4 s	Nie

Więcej warstw = większy koszt obliczeniowy, ale bez poprawy wyników.

# LICZBA WARSTW FC

- Tylko **MaximalFCNN (3 FC)** wykazał spadek straty i oznaki konwergencji.
  - Modele z 1 i 2 warstwami FC **nie uczyły się skutecznie**.
- Zwiększenie liczby FC poprawiło stabilność, ale nie skuteczność klasyfikacji.
  - Brak objawów przeuczenia – nawet przy trzech warstwach.

Hipoteza	Treść hipotezy	Weryfikacja
H1	Dodanie jednej warstwy FC poprawi skuteczność klasyfikacji	Obalona częściowo
H2	Zbyt duża liczba warstw FC prowadzi do przeuczenia	Niepotwierdzona
H3	Model z 1 FC będzie szybki, ale o ograniczonej skuteczności	Potwierdzona
H4	Optymalna liczba FC zależy od złożoności danych i głębokości ekstrakcji cech	Potwierdzona częściowo

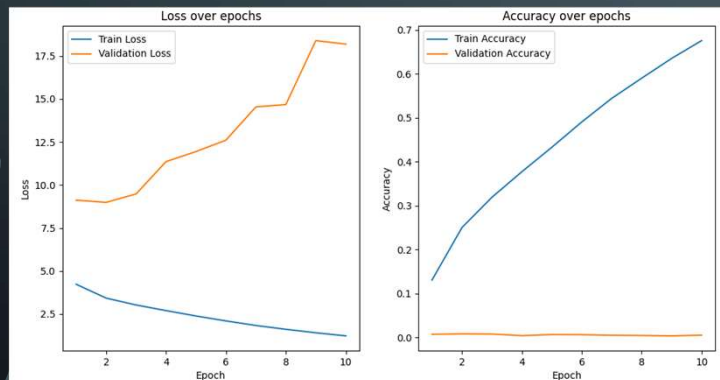


# PRZEUCZENIE A NIEDOUUCZENIE – ANALIZA STABILNOŚCI UCZENIA

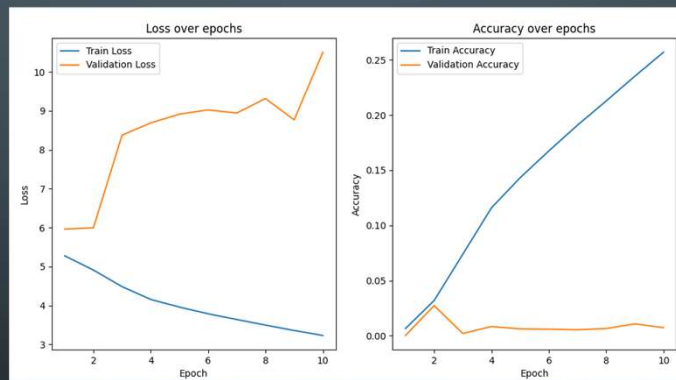
Model	Warstwy konwolucyjne	Warstwy FC	Cechy konstrukcyjne
<b>VerySmallCNN</b>	2	1	Minimalna głębokość
<b>MediumCNN</b>	6	2	Zrównoważona architektura
<b>VeryLargeCNN</b>	18	3	Bardzo głęboka sieć bez stabilizacji

W celu zbadania przeuczenia i niedouczenia przygotowano trzy modele o różnych głębokościach i liczbie warstw w pełni połączonych.

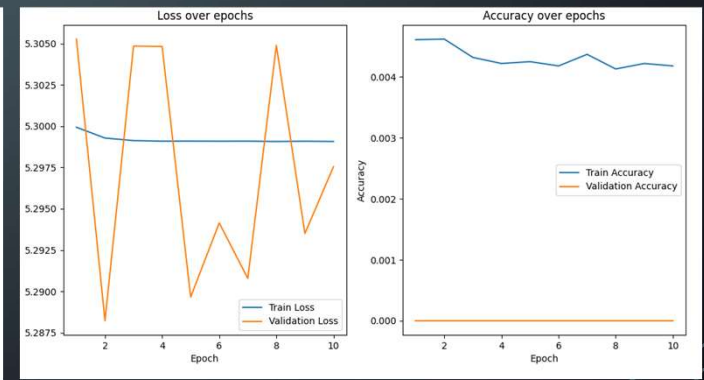
- **VerySmallCNN** – przykład niedouczenia.
- **VeryLargeCNN** – przykład przeuczenia.
- **MediumCNN** – największa stabilność i względna równowaga między dopasowaniem a generalizacją.



VerySmallCNN – wykres strat i dokładności



MediumCNN – przebieg uczenia



VeryLargeCNN – objawy przeuczenia

# PORÓWNANIE WYBRANYCH FUNKCJI AKTYWACJI

**ReLU i Leaky ReLU** – szybkie i stabilne; rekomendowane do dalszego użycia (z regularyzacją).

**Swish i ELU** – potencjalnie wartę ponownego testu w bardziej złożonych modelach.

**Sigmoid i Tanh** – nieskuteczne, problemy z gradientem; nie będą stosowane.

Funkcja aktywacji	Train Loss	Train Acc	Val Loss	Val Acc	Czas treningu (średni)
ReLU	0.3250	90.15%	27.78	0.48%	~35.7s
Leaky ReLU	0.2779	91.94%	26.83	0.71%	~34.6s
ELU	0.3559	90.78%	31.25	0.60%	~34.5s
Swish (SiLU)	0.1897	94.12%	33.42	0.43%	~35.5s
Sigmoid	5.38	0.50%	4.65	0.00%	~34.9s
Tanh	0.3469	89.84%	17.57	0.46%	~34.5s

## ETAP II BADAŃ- EKSPERYMENTY PARAMETRYCZNE

Na podstawie wyników z etapu I przygotowano trzy modele:

**BalancedCNN, UnderfittingCNN i OverfittingCNN,**

reprezentujące różne profile działania sieci — od niedouczenia po przeuczenie.

**Cel:** Zbadanie wpływu parametrów treningowych (learning rate, batch size, optymalizator) na stabilność procesu uczenia, skuteczność klasyfikacji i zdolność generalizacji modeli.

# MODELE REFERENCYJNE

- **BalancedCNN**

Zrównoważona architektura: 6 warstw konwolucyjnych + 2 FC.

Punkt odniesienia w analizie parametrów.

- **OverfittingCNN**

Głęboka sieć (12 Conv + 3 FC), bez regularyzacji.

Stworzona do testów przeuczenia i wpływu LR/optymalizacji.

- **UnderfittingCNN**

Bardzo płytka sieć (2 Conv + 1 FC), niska pojemność.

Służy do analizy niedouczenia i wpływu parametrów.

# WNIOSKI Z DRUGIEGO ETAPU EKSPERYMENTÓW

- **Nie ma uniwersalnych ustawień** – każdy model wymaga indywidualnego dostrojenia.
- **BalancedCNN** – stabilny, ale wrażliwy na dokładny dobór hiperparametrów (najlepszy przy LR 0.0001–0.00001, BS 64).
- **OverfittingCNN** – łatwo się uczył, ale tracił zdolność generalizacji bez regularyzacji.
- **UnderfittingCNN** – niezdolny do efektywnej nauki; zbyt ograniczona architektura.
- **Optymalizatory adaptacyjne** (Adam, RMSprop, AdamW) zapewniały stabilność, ale łatwo prowadziły do przeuczenia bez dodatkowych technik.

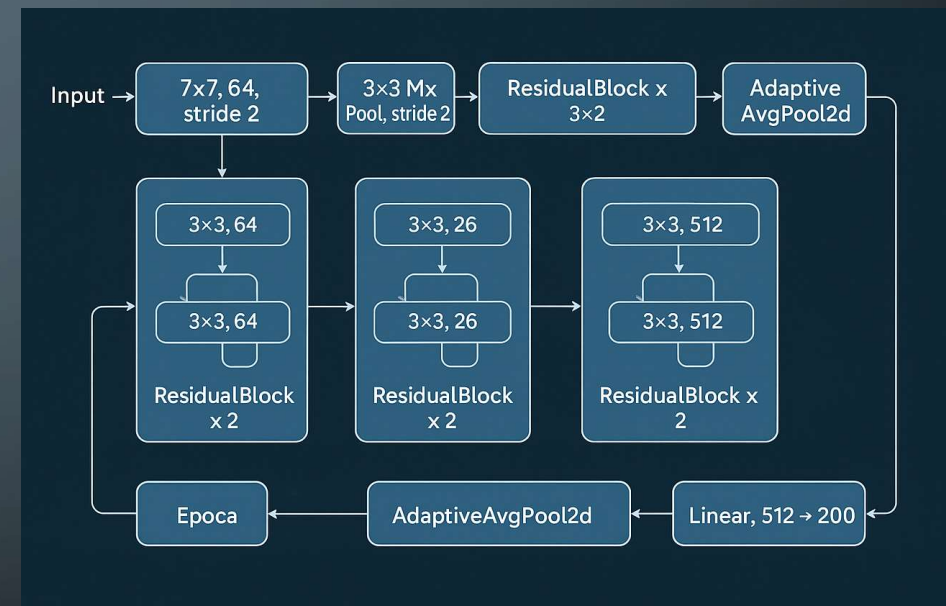
# OPTYMALNE USTAWIENIA – BALANCEDCNN

Parametr	Rekomendowana wartość	Uzasadnienie
<b>Learning rate</b>	0.0001 – 0.00001	Dobre wyniki bez przeuczenia
<b>Batch size</b>	64	Dobry kompromis jakość/czas
<b>Optymalizator</b>	AdamW lub RMSprop	Lepsza stabilność niż SGD
<b>Regularyzacja</b>	Dropout (0.3–0.5) + L2	Ochrona przed przeuczeniem

Zestawienie to stanowi punkt wyjścia do budowy finalnego modelu, opartego na najlepszych parametrach uzyskanych w eksperymentach.

# BUDOWA MODELU KOŃCOWEGO

- Na podstawie wyników eksperymentów i literatury zaprojektowano model o nazwie ResNet18Lite zoptymalizowany do klasyfikacji obrazów z Tiny ImageNet.



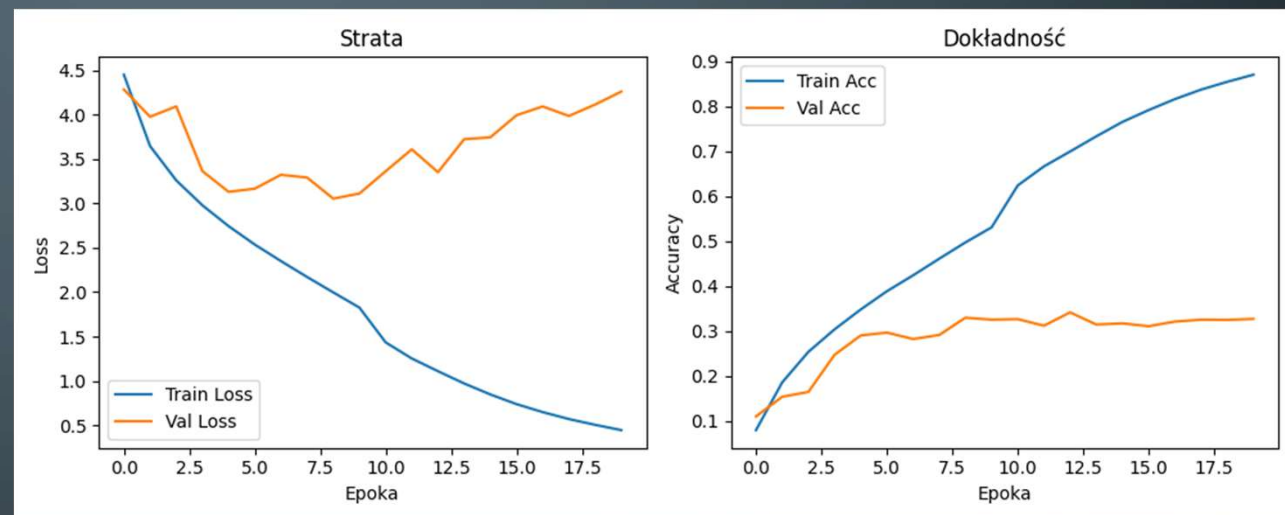


# GŁÓWNE ZAŁOŻENIA KONSTRUKCYJNE

- **18 warstw blokowych** – kompromis między głębokością a stabilnością.
- **ResidualBlock** – lepszy przepływ gradientów, poprawa konwergencji.
- **BatchNorm + ReLU** – sprawdzona para, zwiększa stabilność uczenia.
- **Augmentacja danych** – RandomCrop, Flip, normalizacja → lepsza generalizacja.
- **Parametry treningowe** – **Adam + LR=0.001 + StepLR** → skuteczna i stabilna konfiguracja.

# REZULTATY KOŃCOWEGO MODELU RESNET18LITE

- Kluczowe wyniki:
  - Train acc (epoka 20): ~87%
  - Val acc (epoka 20): ~32.7%
- Przebieg loss/accuracy: stabilny wzrost na treningu, maksimum walidacji ok. epoki 13



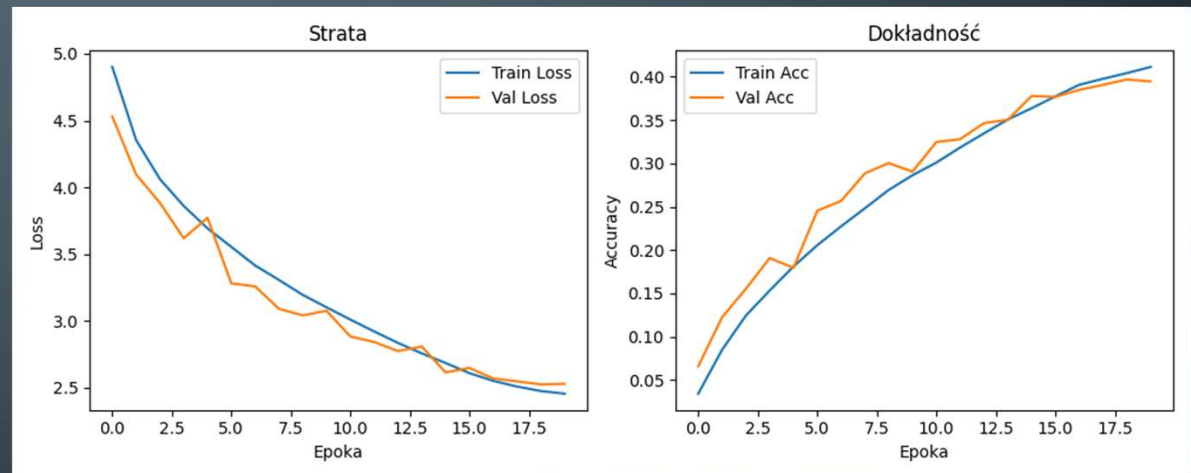
Model osiągnął zadowalającą skuteczność przy niskim koszcie obliczeniowym. Stanowi dobrą bazę do dalszych optymalizacji.

# USPRAWNIENIA ARCHITEKTURY RESNET18LITE

- Najważniejsze zmiany w stosunku do wersji bazowej:
- **Dropout (0.5)** – poprawa generalizacji, redukcja przeuczenia.
- **Zaawansowana augmentacja danych**  
(*RandomResizedCrop, Rotation, ColorJitter, HorizontalFlip*).
- **Nowa strategia uczenia**
  - Optymalizator: **Adam** + *weight decay*
  - Scheduler: **CosineAnnealingLR**

# WYNIKI KOŃCOWE – MODEL ULEPSZONY

- Train Accuracy: 41.1%
- Val Accuracy: ~39.7%
- Val Loss: systematyczny spadek (z 4.53 → 2.53)
- Brak oznak przeuczenia – linie strat zbieżne



Model ResNet18LiteV2 wykazuje zrównoważony trening i wysoką skuteczność generalizacji bez wyraźnych objawów przeuczenia.

# PORÓWNANIE WYNIKÓW Z LITERATURĄ

- **Stanford CS231n (2017):**

ResNet-18 na Tiny ImageNet → **ok. 40%** val accuracy.

- **GitHub (Tiny-ImageNet-Classifler):**

Transfer learning z ImageNet → **25.9%** → **56.9%**.

- **Benchmarki TinyImageNet:**

ResNet-18 (bez TL) osiąga **~41.5%** test accuracy.

Osiągnięta dokładność ~40% w treningu od podstaw jest zgodna z literaturą i potwierdza skuteczność zastosowanych optymalizacji.

# STRATEGIA TESTOWANIA GOTOWYCH SIECI

- *ETAPY TESTOWANIA MODELI PREDEFINIOWANYCH:*

- **Zero-shot (bez uczenia):**

Model oceniany bez treningu. Oczekiwana skuteczność: **bardzo niska**.

- **Feature extraction:**

Uczony tylko klasyfikator. Oczekiwana skuteczność: **średnia**, szybki trening.

- **Fine-tuning:**

Trening całego modelu (Tiny ImageNet + regularyzacja, LR scheduler).

Skuteczność: **najwyższa**, wysoki koszt obliczeniowy.

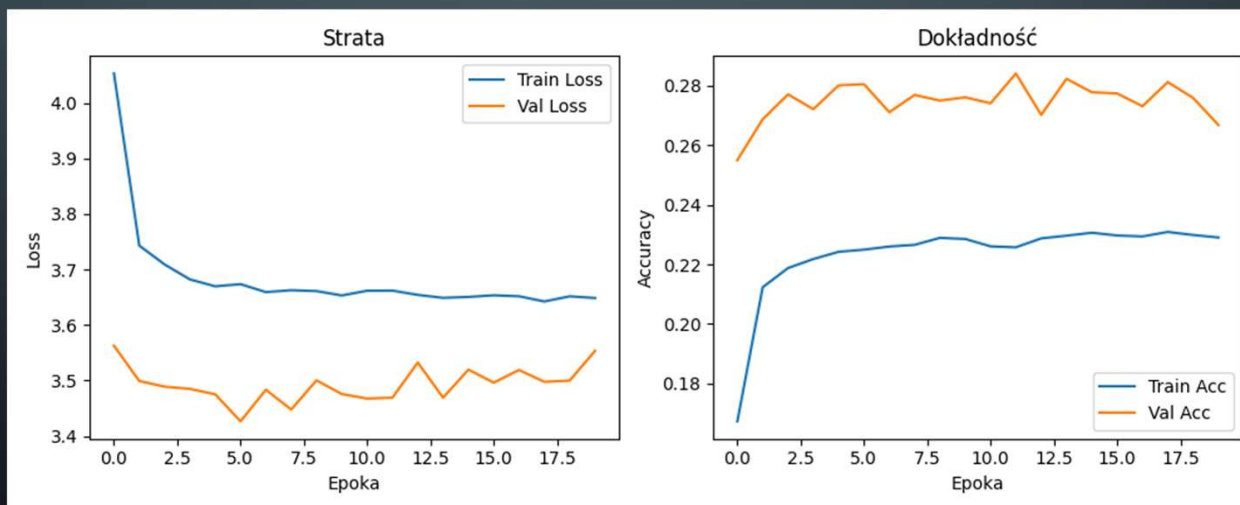
# WYNIKI MODELU RESNET18 – TINY IMAGENET

## *Wyniki testu zero-shot – ResNet18 na Tiny ImageNet*

- Model użyty bez dodatkowego treningu (oryginalna baza i klasyfikator z ImageNet).
- Tiny ImageNet to podzbiór ImageNet, ale wyniki były **skrajnie niskie**.
- **Validation accuracy: 0.46%** – potwierdza brak adaptacji i losową klasyfikację.

# WYNIKI MODELU RESNET18 – TINY IMAGENET

- Tryb **classifier\_train** – trenowano tylko ostatnią warstwę klasyfikującą (reszta zamrożona).
  - Osiągnięta **dokładność walidacyjna: 26.7%**.
  - Przebieg treningu stabilny, **bez przeuczenia**.
- Niski koszt obliczeniowy, dobra skuteczność względem testu zero-shot.

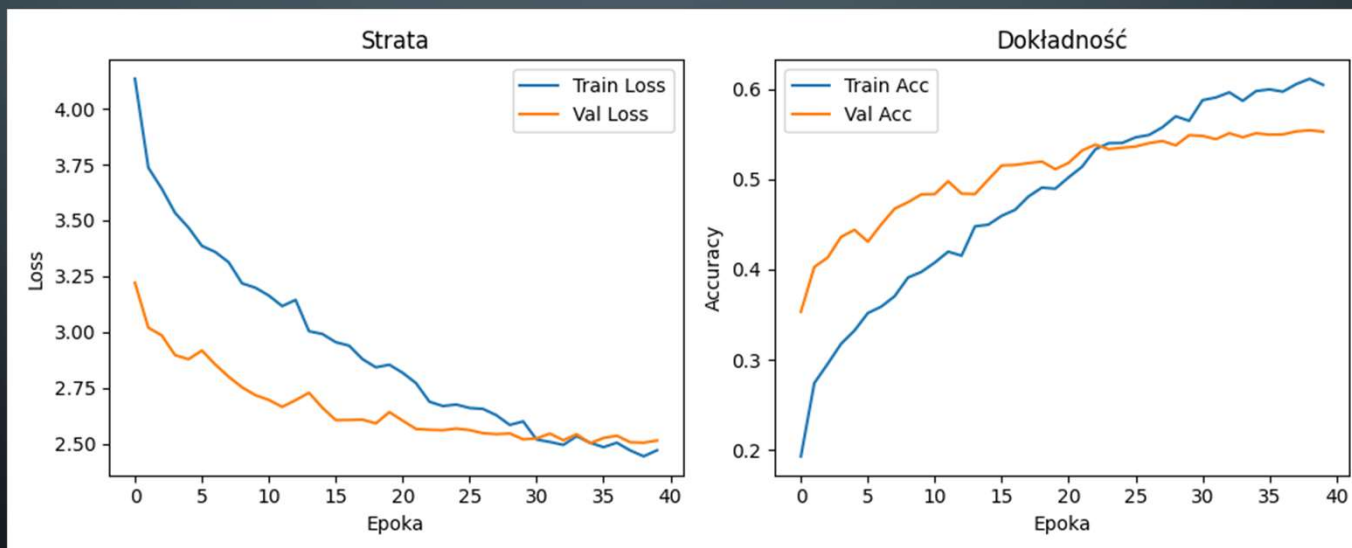




# WYNIKI MODELU RESNET18 – TINY IMAGENET

*Tryb: Fine-tuning (pełne dostosowanie)*

- Wszystkie warstwy trenowane przez 40 epok.
- Najlepszy wynik spośród wszystkich podejść:  
**Val accuracy: 55.3%, F1-score: ~54.8%**
- Brak przeuczenia, stabilny przebieg uczenia.
- Najwyższa jakość, kosztem większej złożoności.



# ETAP PRACY

## Zrealizowano:

- Eksperymenty wstępne
- Analiza parametrów treningowych na trzech modelach
  - Budowa i test własnego modelu (~40%)
  - Porównanie gotowych modeli w trzech trybach

## W toku:

- Opis własnego modelu i jego wyników
- Redagowanie części eksperymentalnej i wniosków

An abstract graphic on the left side of the slide, consisting of a network of thin, light blue lines and small circles, resembling a circuit board or a neural network diagram. The lines and circles are arranged in a vertical, branching pattern, with some lines extending horizontally and others at angles. The circles are of varying sizes and are placed at various points along the lines.

DZIĘKUJĘ ZA UWAGĘ