RAPORT - TESTOWANIE I OPTYMALIZACJA KODU

1. Dane.

Whole set	Train samples		Validation samples	Test samples
15114	***ALL***	***HEM***	1867	2586
	7272	3389		
	10661			

2. Porównanie CPU vs GPU vs GPU + transfer learning.

Dla wstępnego porównania CPU vs GPU vs GPU + Transfer Learning użyto ograniczonego zbioru: **1 000 obrazów treningowych, 300 walidacyjnych** — aby przyspieszyć testy CPU.

COMPARING GPU vs CPU

	Model	Training time (min)	Epochs	Train/Validation samples	Accelerator	Transfer learning	Normalisation	Augmentation	Dropout	New data samples	Input size	Batch size	Train accuracy	Validation accuracy
1	ResNet50	83.64	10	1000/300	-	-	-	-	-	-	224x224	32	0.84	0.33
2	ResNet50	3.22	10	1000/300	GPU	-	-	-	-	-	224x224	32	0.842	0.66
3	**ResNet50**	**1.01**	**10**	**1000/300**	**GPU**	**+**	**-**	**-**	**-**	**-**	**224x224**	**32**	**0.841**	**0.6667**

Wnioski:

- Przejście z CPU na GPU znacznie skróciło czas treningu: z **84 min do ok. 1–3 min**.
- Włączenie **transfer learningu (na GPU)** dodatkowo ustabilizowało wyniki i jeszcze bardziej skróciło czas.
- Brak normalizacji, augmentacji i dropout'u w tym etapie skutkował lekkim overfittingiem, ale GPU + TL częściowo ten problem ograniczyło.

GPU + transfer learning wypada tutaj najlepiej.

3. Optymalizacja (p. 4 wymagania projektu).

Na podstawie wyników z wstępnych testów, wybrano:

- **GPU + Transfer Learning**
- **Input size: 224x224**
- **Batch size = 32**

W kolejnych iteracjach:

- Użyto całego zbioru danych (10 661 obrazów treningowych i 1 867 walidacyjnych).
- Dodano stopniowo **normalizację, augmentację** (zbalansowanie klas — by obie klasy ALL i HEM miały po 7 272 zdjęcia, aby zapobiec bias).

- Wprowadzono **dropout 0.3.**

OPTIMALISATION

	Model	Training time (min)	Epochs	Train/Validation samples	Accelerator	Transfer learning	Normalisation	Augmentation	Dropout	New data samples	Input size	Batch size	Train accuracy	Validation accuracy		
4	ResNet50	9.63	10	10661/1867	GPU	+	-	-	-	+	224x224	32	0.841	0.6667	4d	<- from here all optimisation techniques will use 10661/1867 set
5	ResNet50	10.26	10	10661/1867	GPU	+	+	-	-	-	224x224	32	0.8427	0.6952	4a	<- normalization alone (normalised whole dataset)
6	ResNet50	9.49	10	14544/1867	GPU	+	-	+	-	+	224x224	32	0.8914	0.699	4b.1	<- augmentation alone (only for 'HEM') so the model isn't biased
7	ResNet50	11.08	10	14544/1867	GPU	+	+	+	-	-	224x224	32	0.8862	0.6883	4b.2	<- augmentation (-|-) + normalisation
8	ResNet50	11.56	10	14544/1867	GPU	+	+	+	included: 0.3	-	224x224	32	0.8698	0.6824	4c	<- dropout with augmentation + normalisation
9	ResNet50	8.39	10	14544/1867	GPU	+	+	+	included: 0.3	-	96x96	32	0.8656	0.6845	4e.1	<- 96x96, 160x160, 224x224 input size
10	ResNet50	9.41	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	32	0.8675	0.6872	4e.2/4f (batch_size = 32)	
11	ResNet50	9.58	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	64	0.8698	0.6797	4f.1	<- 32/65/128 batch_size
12	ResNet50	11.1	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	128	0.8639	0.6754	4f.2	

Wnioski:

- **Normalizacja i augmentacja (szczególnie zbalansowanie klas)** mają największy wpływ na poprawę dokładności walidacyjnej.
- **Dropout** skutecznie ogranicza overfitting.
- **Zbyt duży batch size** obniża zdolność uogólniania i wydłuża czas obliczeń.
- **Rozmiar wejściowy ok. 160x160** jest najbardziej optymalny — kompromis między jakością predykcji a czasem treningu.

e) różne rozmiary wejściowe (224x224, 96x96, 160x160)

InputSize comparison

Model	Training time (min)	Epochs	Train/Validation samples	Accelerator	Transfer learning	Normalisation	Augmentation	Dropout	New data samples	Input size	Batch size	Train accuracy	Validation accuracy	
ResNet50	11.56	10	14544/1867	GPU	+	+	+	included: 0.3	-	224x224	32	0.8698	0.6824	
ResNet50	**8.39**	10	14544/1867	GPU	+	+	+	included: 0.3	-	96x96	32	0.8656	0.6845	
ResNet50	**9.41**	**10**	**14544/1867**	**GPU**	+	+	+	**included: 0.3**	-	**160x160**	**32**	**0.8675**	**0.6872**	<- best overall

Wnioski:

- Dla **ResNet50** mniejsze rozdzielczości (96x96, 160x160) skracają czas obliczeń bez pogorszenia jakości — a nawet delikatnie poprawiają dokładność walidacyjną (val_acc).
- **160x160**: najlepsza generalizacja + krótki czas treningu.

f) różny rozmiar Batch size (32, 64, 128)

BatchSize comparison

Model	Training time (min)	Epochs	Train/Validation samples	Accelerator	Transfer learning	Normalisation	Augmentation	Dropout	New data samples	Input size	Batch size	Train accuracy	Validation accuracy	
ResNet50	**9.41**	**10**	**14544/1867**	**GPU**	+	+	+	**included: 0.3**	-	**160x160**	**32**	**0.8675**	**0.6872**	<- most balanced
ResNet50	9.58	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	64	0.8698	0.6797	
ResNet50	11.1	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	128	0.8639	0.6754	

Wnioski:

- **Batch size 32** daje najlepszy balans między dokładnością a czasem treningu.
- **Batch size powyżej 64 (np. 128)** powoduje spadek dokładności walidacyjnej (val_acc) i dłuższy czas obliczeń.

g) różne struktury sieci (InceptionV3, ResNet101, DenseNet121, EfficientNetB0)

Different models (took 'best' parameters based on training time and train&validation accuracy)

Model	Training time (min)	Epochs	Train/Validation samples	Accelerator	Transfer learning	Normalisation	Augmentation	Dropout	New data samples	Input size	Batch size	Train accuracy	Validation accuracy	
InceptionV3	10.49	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	32	0.7967	0.661	
ResNet101	**10.75**	**10**	**14544/1867**	**GPU**	+	+	+	**included: 0.3**	-	**160x160**	**32**	**0.8647**	**0.7022**	<- best overall
DenseNet121	10.67	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	32	0.8302	0.6786	
EfficientNetB0	10.39	10	14544/1867	GPU	+	+	+	included: 0.3	-	160x160	32	0.827	0.6743	

Wnioski:

- **ResNet101** osiągnął **najwyższe val_acc (0.7022)** — najlepsza jakość predykcji.

- **InceptionV3** i **EfficientNetB0** są nieco *szybsze*, ale mają wyraźnie niższą dokładność.
- **DenseNet121** to kompromis, ale nie ma przewagi nad ResNet101.

4. Wnioski końcowe:

Najlepsza konfiguracja:

- Architektura: ResNet101
- Input size: 160x160
- Batch size: 32
- GPU + Transfer Learning
- Normalizacja, augmentacja (zbalansowanie klas HEM, ALL), dropout 0.3