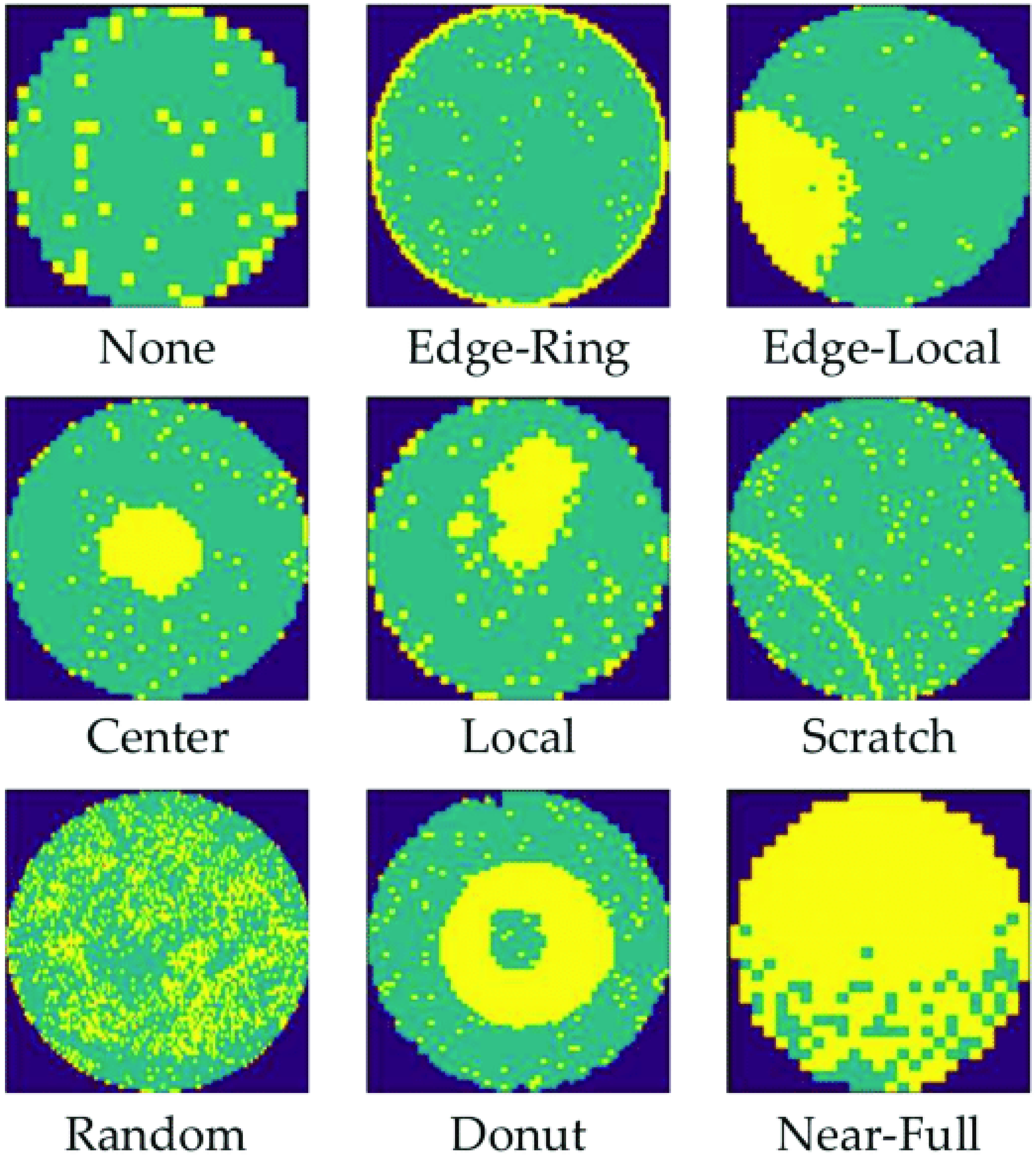


WM-811K WaferMap

Model evaluation for semiconductor wafermap detection and recognition using CNNs

Data Analysis
The problem

- **Classification task**:
  - Semiconductor wafermap.
  - 9 different anomalous classes: none, loc, edge-loc, center, donut, edge-ring, near-full, random, scratch.
  - **CNNs**.

- **WM-811K dataset**:
  - 811457 images (172950 labelled).
  - Very **unbalanced**.
  - **3** kinds of **pixel**: 0 for background, 1 for regular chips, 2 for anomalies.

Data Analysis
Preprocessing

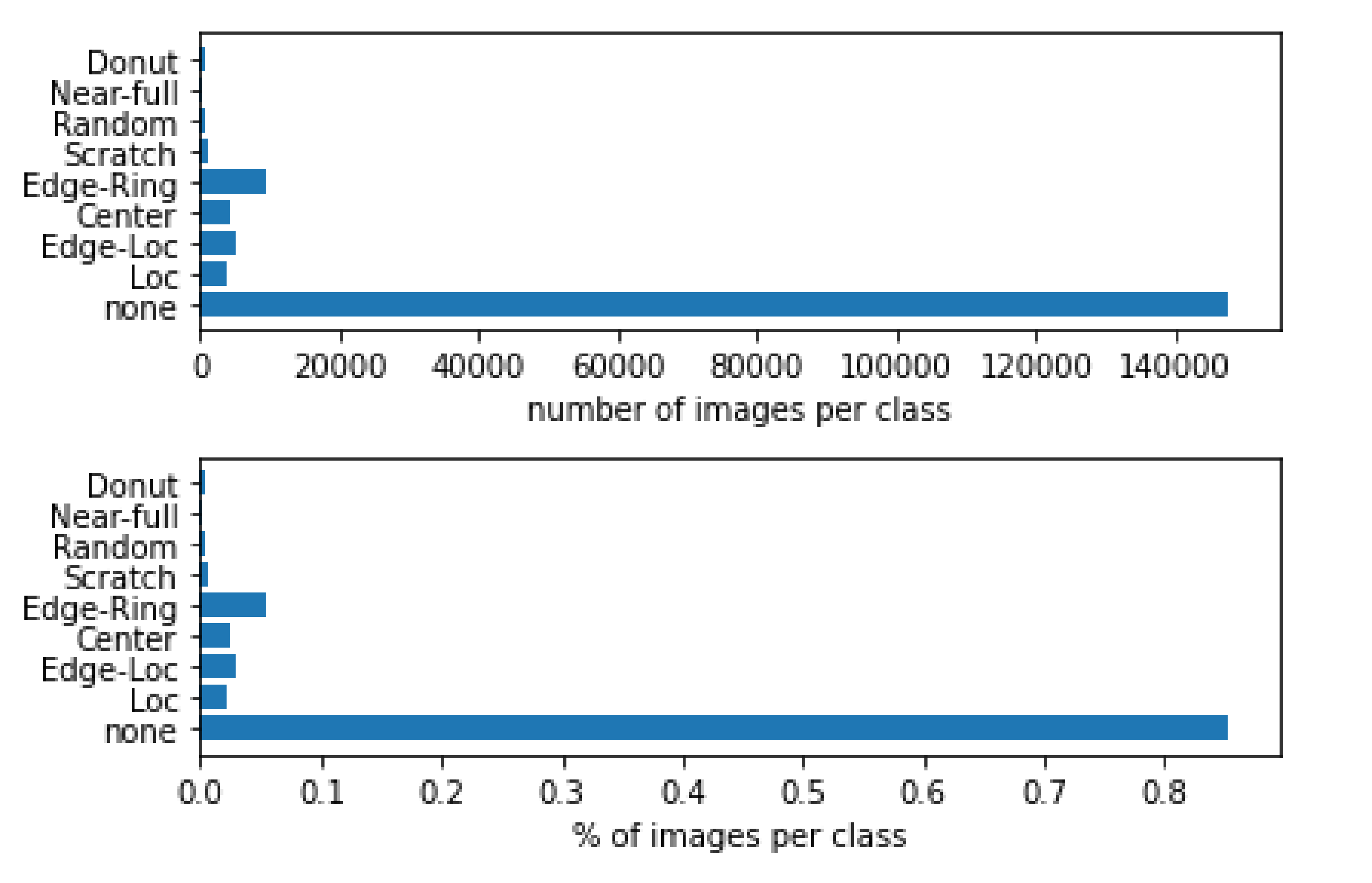


Fig.1: Image distribution over classes

- We only consider **labelled** images.
- Extreme **imbalance** towards none class (no anomalies) and towards singular anomalous classes.
- 346 different images **shapes**.
- Median shape: **(53, 52)**.
- None class associated also to **defective** images.

Data Analysis
Resize and cleaning images

- **Resize** images to the **median** shape: (53, 52).
- **Remove** images with a strongly **rectangular** shape.
- **Remove defective** images.
- Split **train-test** (80%-20%).
- **Rebalance** with **data augmentation** (only on training-set).

Fig.2: Defective images

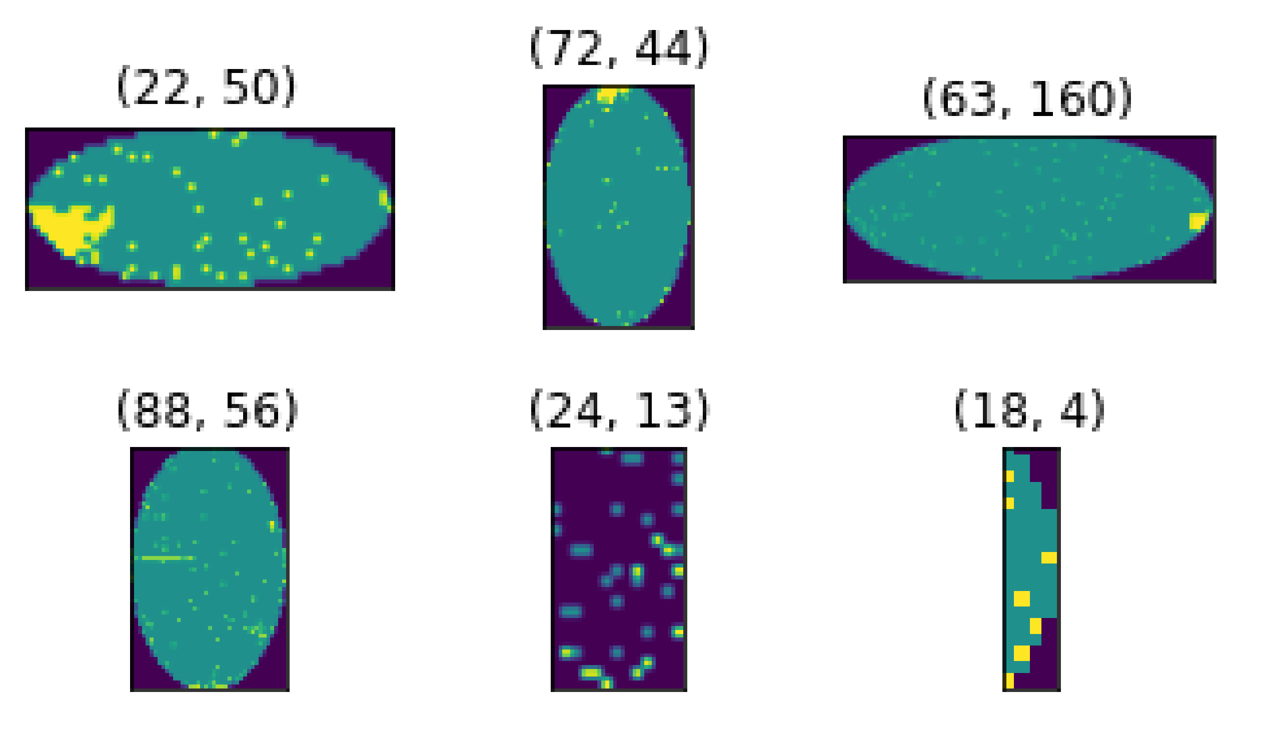


Fig.3: Rectangular shapes

Data Augmentation
Craft-handed

- Before rebalancing the "none" class, we need to **rebalance** the **anomalous** classes themselves.

- **Idea**: Take an image with no defective chips as **template** and change the pixels depending on the anomaly class to be increased.

- At least **3000** images for each class: augment **donut**, **near-full**, **random**, **scratch** and **loc** anomalies.

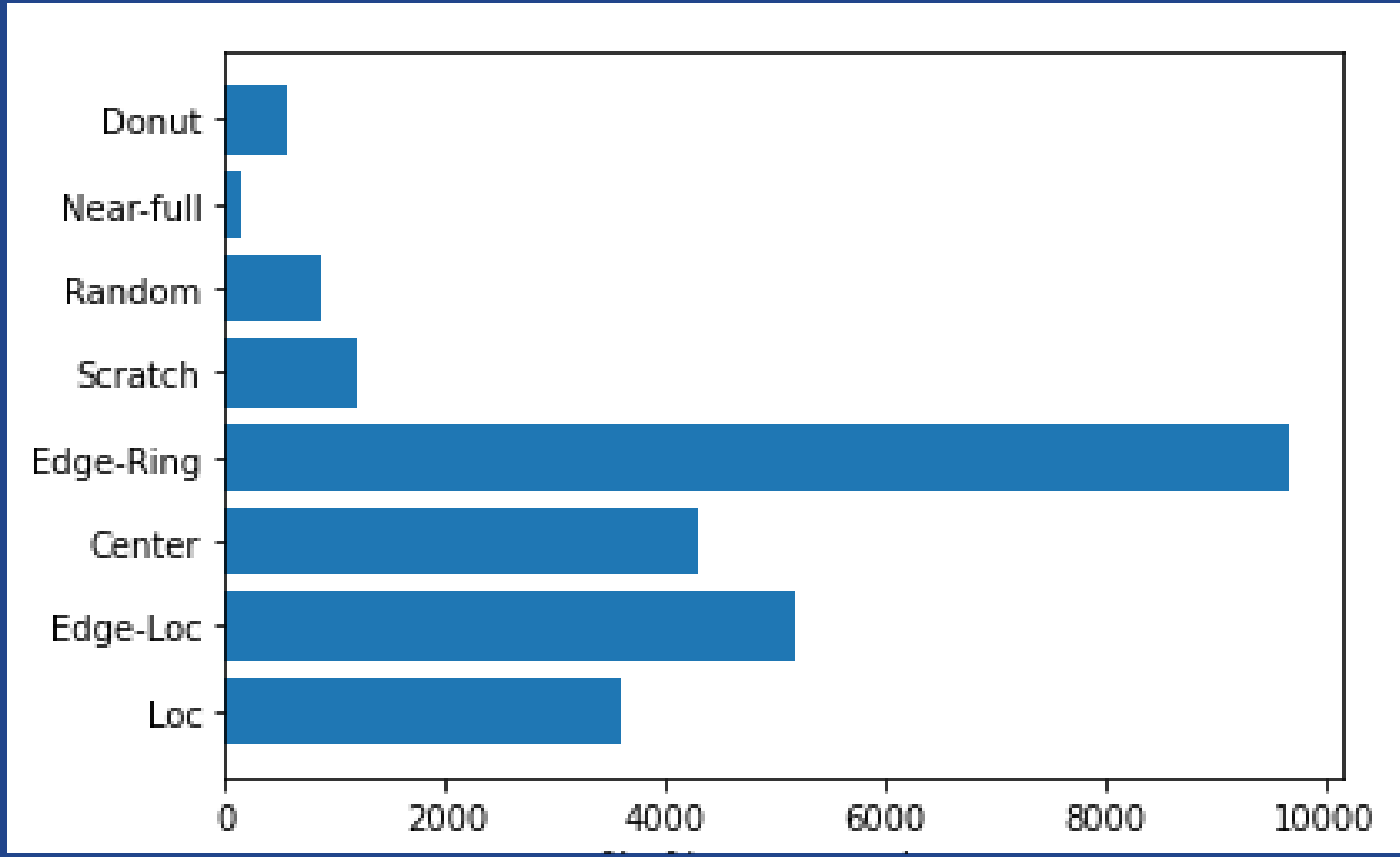


Fig.4: Unbalance anomalies

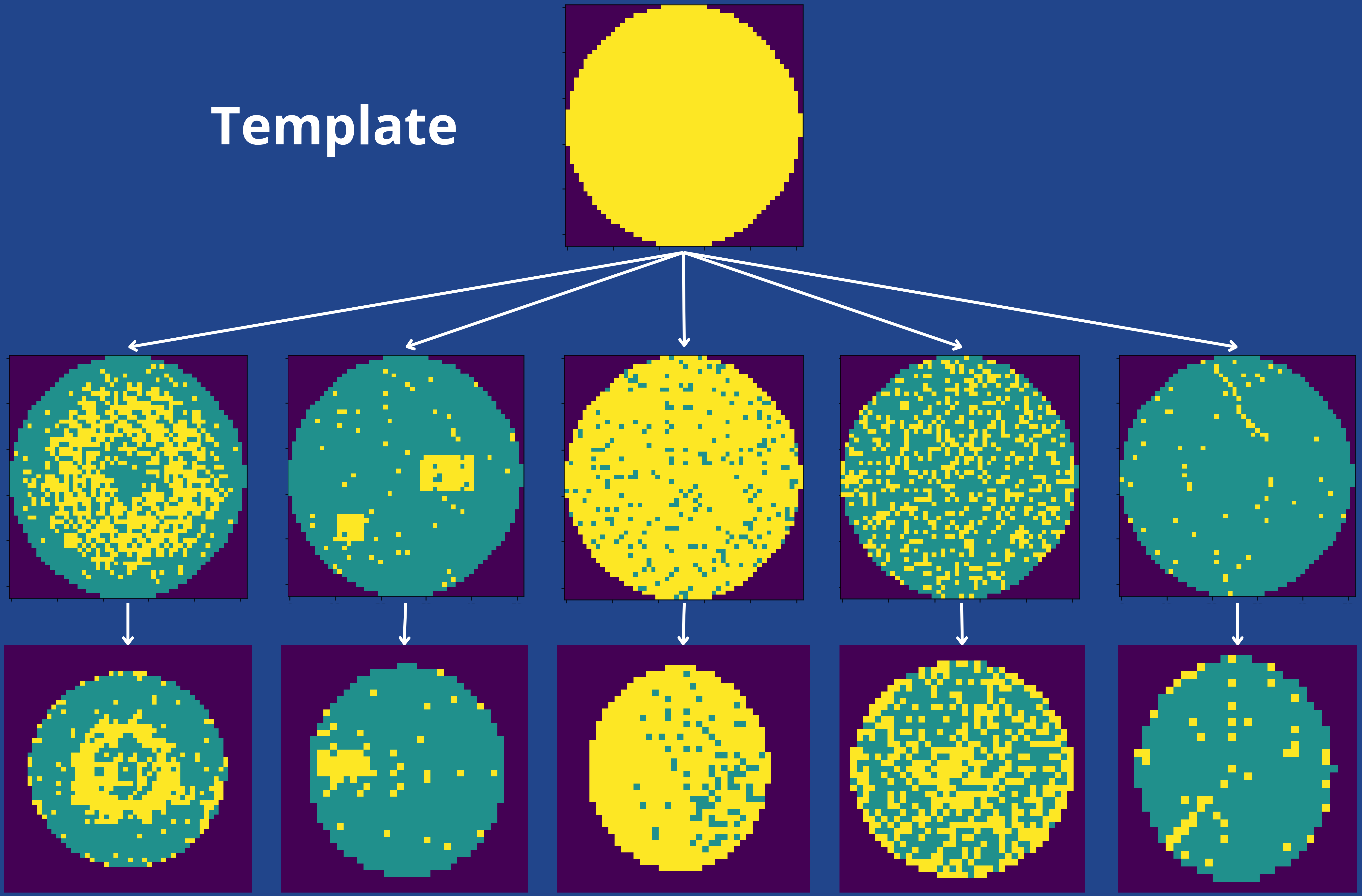
Template

Craft-handed images

(accuracy 84.13%)

Real images

(accuracy 88%)



Donut | Loc | Near-full | Random | Scratch

Data Augmentation
Flipping + rotation

- **For each** anomalous image:
  - Horizontal and vertical **flipping**.
  - **Rotation** (90° and 270°).

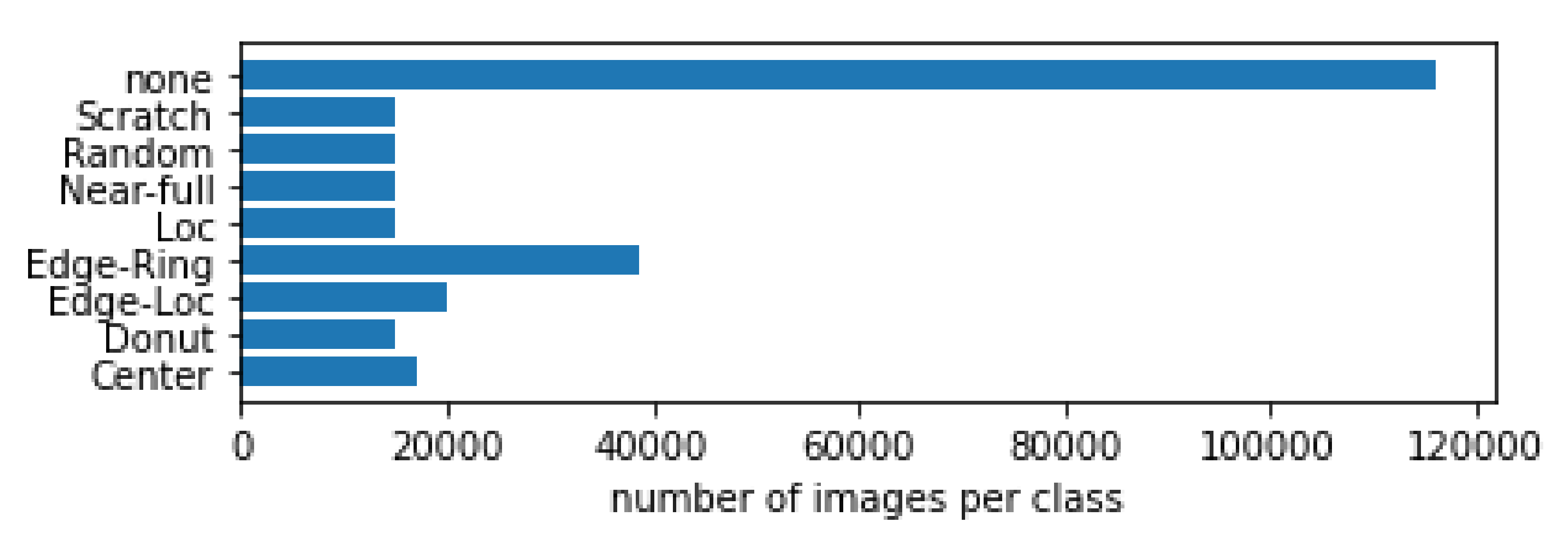


Fig.5: Rebalancing classes

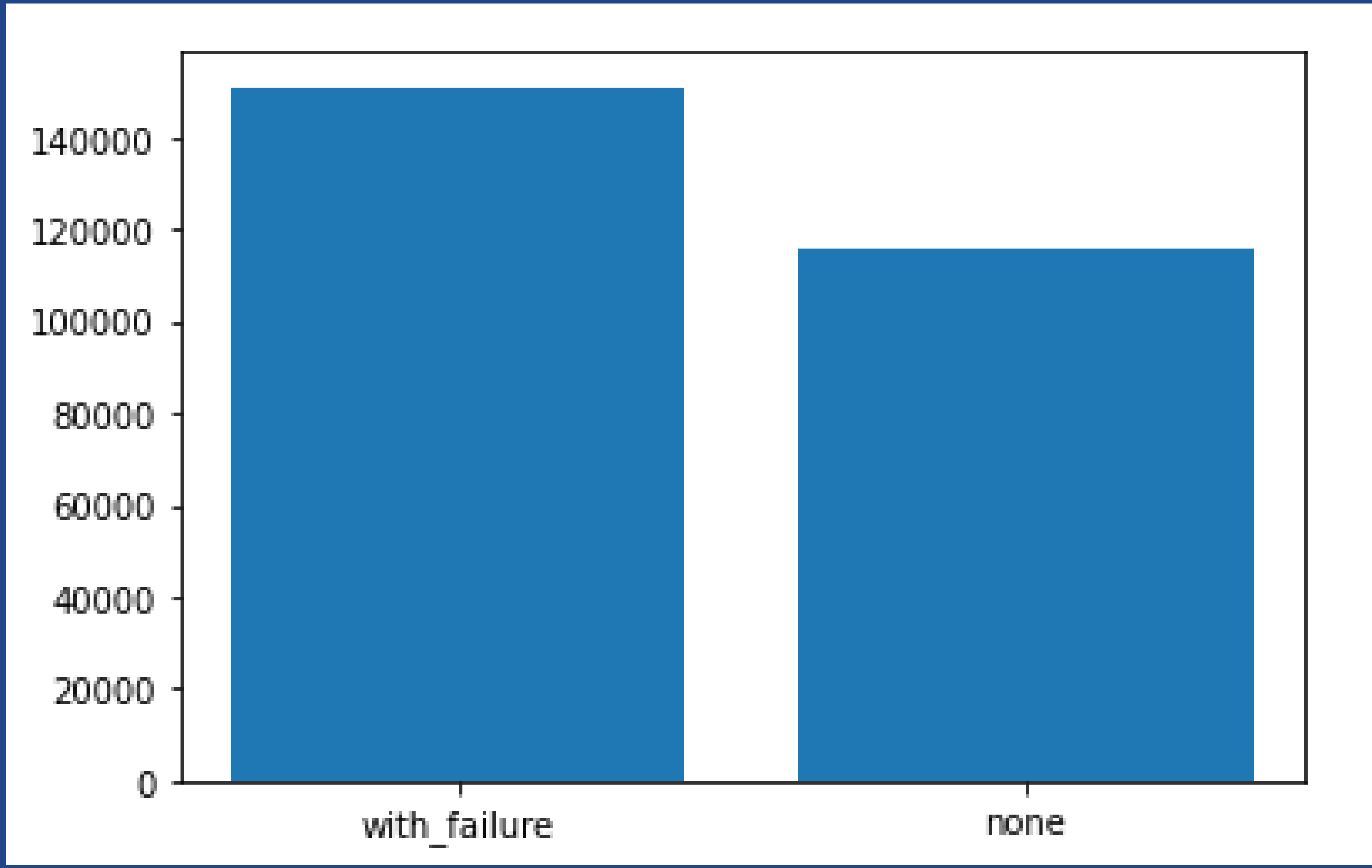


Fig.6: Rebalancing anomalies vs none

- 136196 ⟶ 266947 training images.

Model selection

Base model

- **3 convolutional** layers:
  - 32, 64 and 128 filters.
  - receptive field = 3.
  - padding = same.
  - activation = **ReLU**.
- **3 pooling** layers:
  - pooling window 3x3.
  - padding = same.
- **1 fully-connected** layer:
  - units = 9.
  - activation = **Softmax**.

- **Compile**:
  - loss = categorical_crossentropy.
  - optimizer = **Adam** (**learning rate = 0.001**).
  - metrics = accuracy.

- Fitting for **20 epochs separately** on training data "vanilla" and training data augmented.

- Various models tested. Here we show the **4 main ones and the final model**.

- Comparisons based on the previous best model.

"Vanilla" data - Model 0

Fig.7: Base model - training "vanilla"

Best **val_loss**: 0.1155
Best **val_accuracy**: 96.75%

Precision: 95.51%
Recall: 88.13%
F1-measure: 91.67%

	Model 0
Center	88.61%
Donut	79.78%
Edge-Loc	66.46%
Edge-Ring	97.93%
Loc	66.61%
Near-full	79.17%
Random	84.78%
Scratch	29.03%
None	99.33%
Mean	76.86%

Augmented data - Model 01

Fig.8: Base model - training augmented

Best **val_loss**: 0.1095 (-0.0060)

Best **val_accuracy**: 96.88% (+0.13%)

Precision: 87.77%

Recall: 92.86%

F1-measure: 90.25% (-1.42%)

	Model 0	**Model 01**
Center	88.61%	91.09%
Donut	79.78%	76.40%
Edge-Loc	66.46%	79.70%
Edge-Ring	97.93%	98.19%
Loc	66.61%	74.96%
Near-full	79.17%	100%
Random	84.78%	82.61%
Scratch	29.03%	39.25%
None	99.33%	97.91%
Mean	76.86%	82.34%

BatchNormalization - Model 3

- EarlyStopping (epochs = 100, patience = 10)
- He-initialization
- Learning rate = 0.0001
- **BatchNormalization** layer

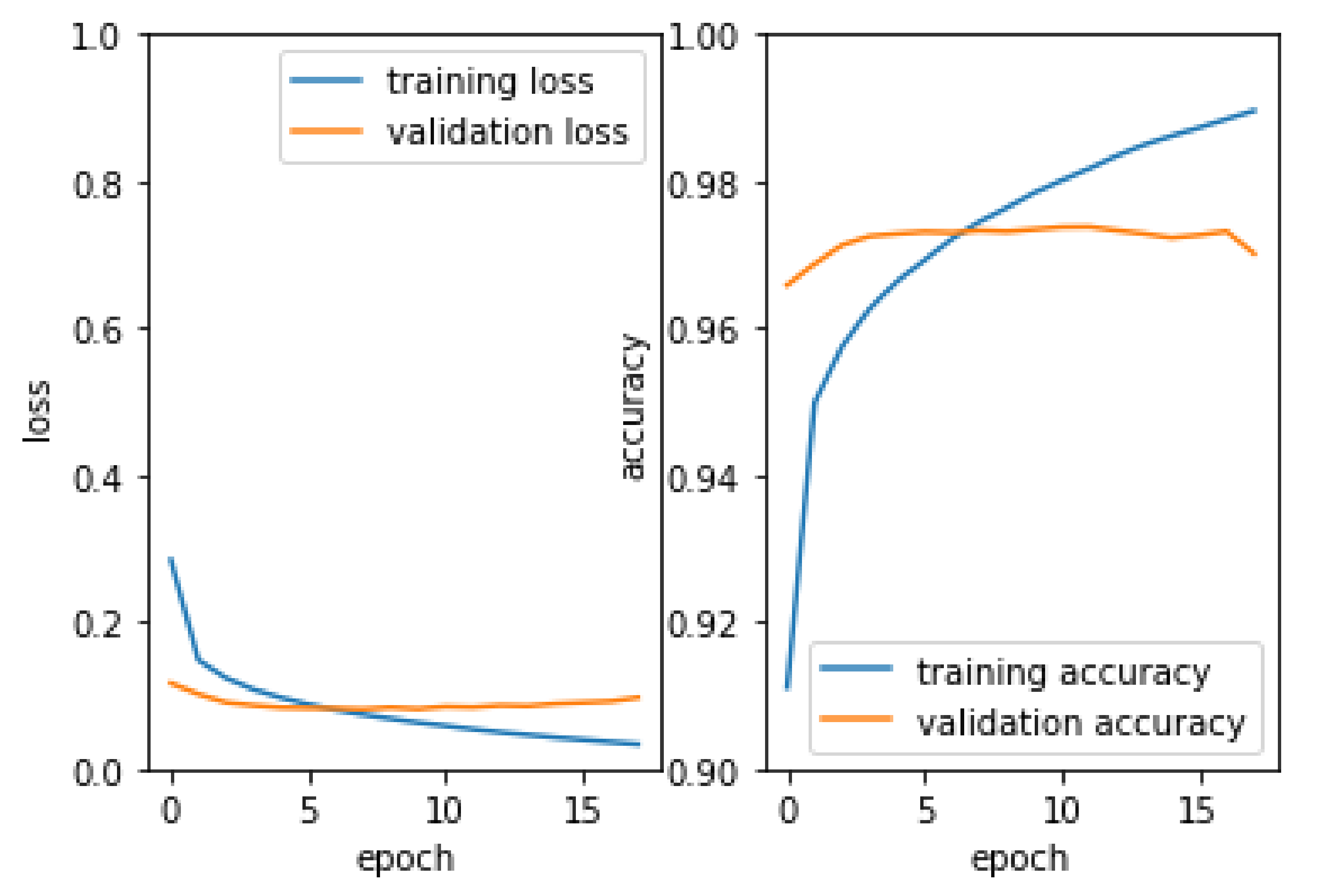


Fig.9: BatchNormalization

Best **val_loss**: 0.0808 (-0.0287)

Best **val_accuracy**: 97.38% (+0.5%)

Precision: 91.51%

Recall: 93.97%

F1-measure: 92.72% (+2.47%)

	Model 0	Model 01	Model 3
Center	88.61%	91.09%	88.19%
Donut	79.78%	76.40%	91.01%
Edge-Loc	66.46%	79.70%	84.16%
Edge-Ring	97.93%	98.19%	97.48%
Loc	66.61%	74.96%	78.72%
Near-full	79.17%	100%	95.83%
Random	84.78%	82.61%	81.88%
Scratch	29.03%	39.25%	55.14%
None	99.33%	97.91%	98.58%
Mean	76.86%	82.34%	85.67%

Decreasing complexity - Model 6

- EarlyStopping (epochs = 100, patience = 10)
- He-initialization
- Learning rate = 0.0001
- BatchNormalization layer
- Conv layer with **16, 32 and 64 filters** + pooling window **2x2**

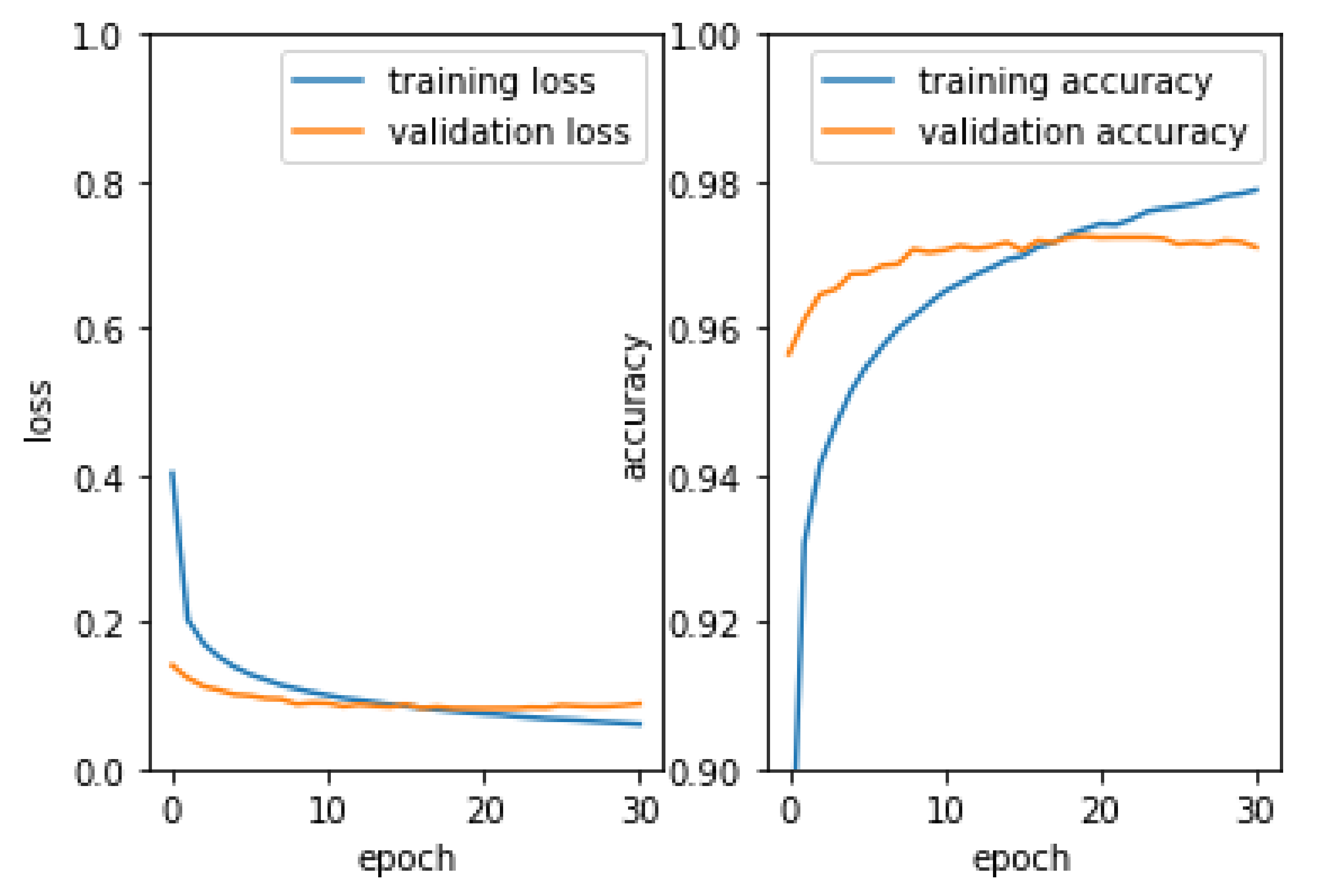


Fig.10: Decreasing complexity

Best **val_loss**:	0.0819 (+ 0.011)	**Precision**:	92.69%
		Recall:	94.28%
Best **val_accuracy**:	97.25% (-0.13%)	**F1-measure**:	93.48% (+0.76%)

	Model 01	**Model 3**	**Model 6**
Center	91.09%	88.19%	89.65%
Donut	76.40%	91.01%	86.52%
Edge-Loc	79.70%	84.16%	87.00%
Edge-Ring	98.19%	97.48%	96.64%
Loc	74.96%	78.72%	64.72%
Near-full	100%	95.83%	91.67%
Random	82.61%	81.88%	90.58%
Scratch	39.25%	55.14%	67.57%
None	97.91%	98.58%	98.8%
Mean	82.34%	85.67%	85.9%

Model selection
Parameters tuning

- Starting point : Model 6 (**decreasing complexity**).

- Searching the best combination of **learning rate** and **L2 regularization**.

- **KerasTuner**:
  - Objective: **val_loss**.
  - Learning rate = [0.0001, 0.001].
  - L2 regularization = [0.0, 1e-7, 1e-6, 1e-5].
  - Training each model for **10 epochs**.

- Results: **learning rate = 0.0001** and **L2 = 0.0**, i.e. **Model 6**.

Performance evaluation

Model6

	Loss	Top1 Accuracy	Top3 Accuracy	Top5 Accuracy	Precision	Recall	F1-measure	Mean over single classes
Test	0.0884	97.07%	99.83%	99.99%	92.25%	93.58%	92.91%	85.67%
Validation	0.0882	97.10%	99.82%	99.99%	92.69%	94.28%	93.48%	85.90%

Training	0.0608	97.88%

Gravina Greta (ID 881470)
Puccinelli Niccolò (ID 881395)
Scatassi Marco (ID 883823)

Thanks for your attention

g.gravina8@campus.unimib.it
n.puccinelli@campus.unimib.it
m.scatassi@campus.unimib.it