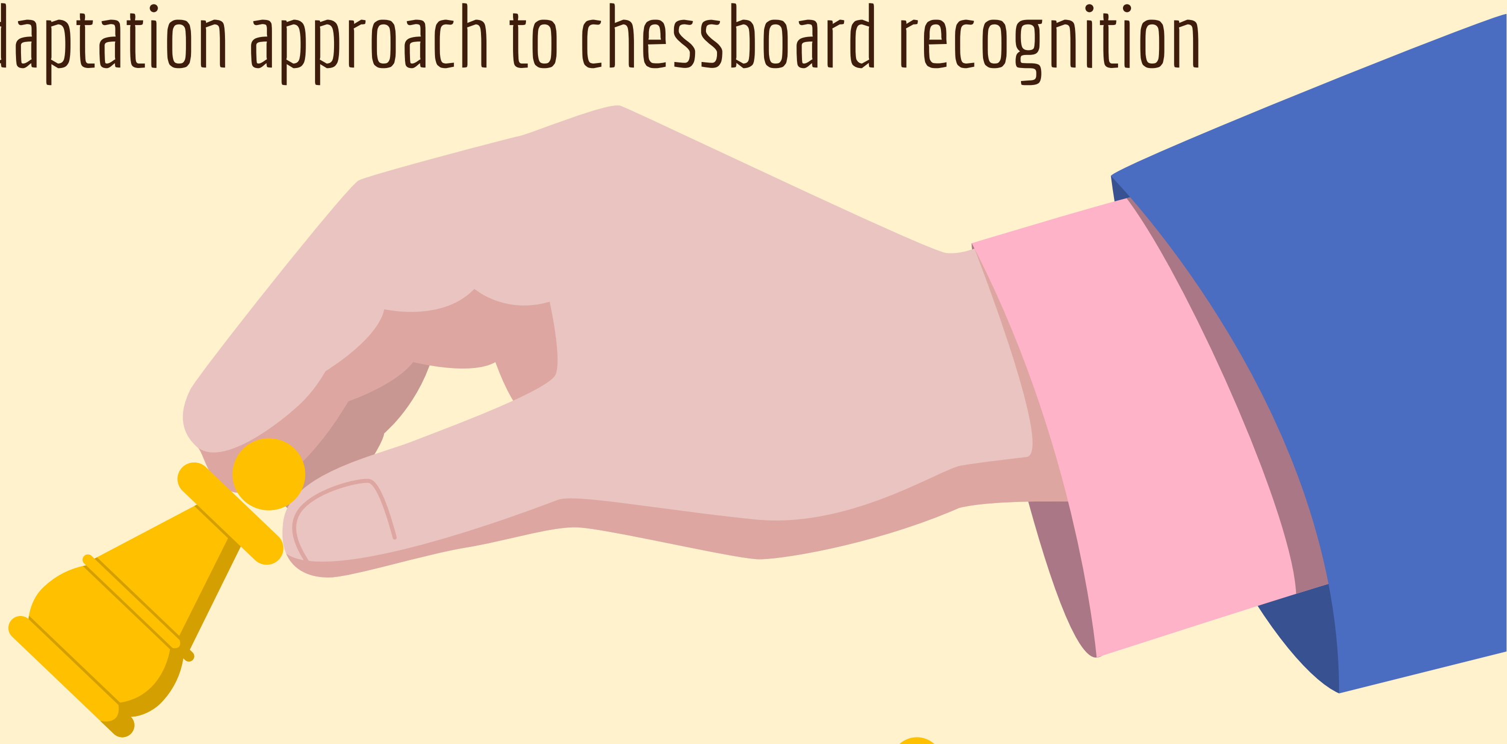
RecogniChess

An unsupervised domain-adaptation approach to chessboard recognition



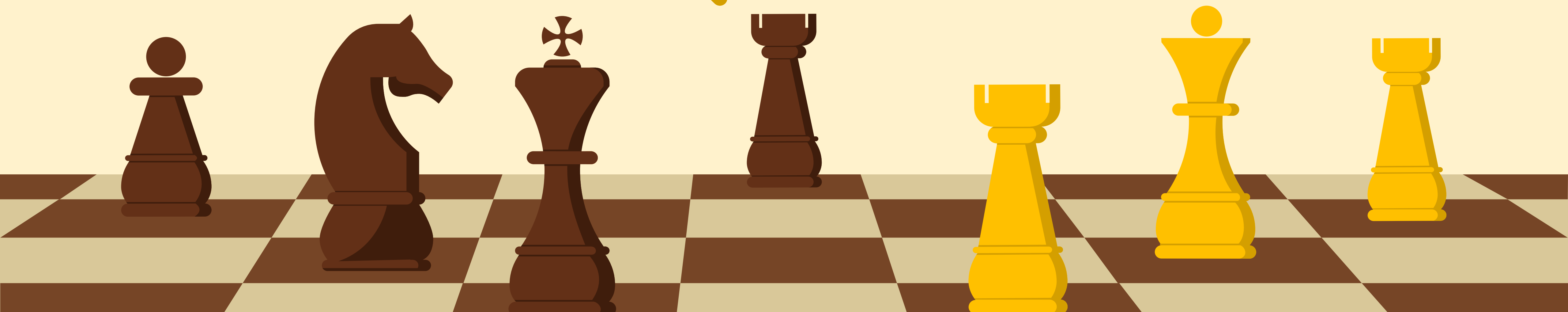
Group 10

Wassim Jabbour - 260969699

Enzo Benoit-Jeannin - 260969262

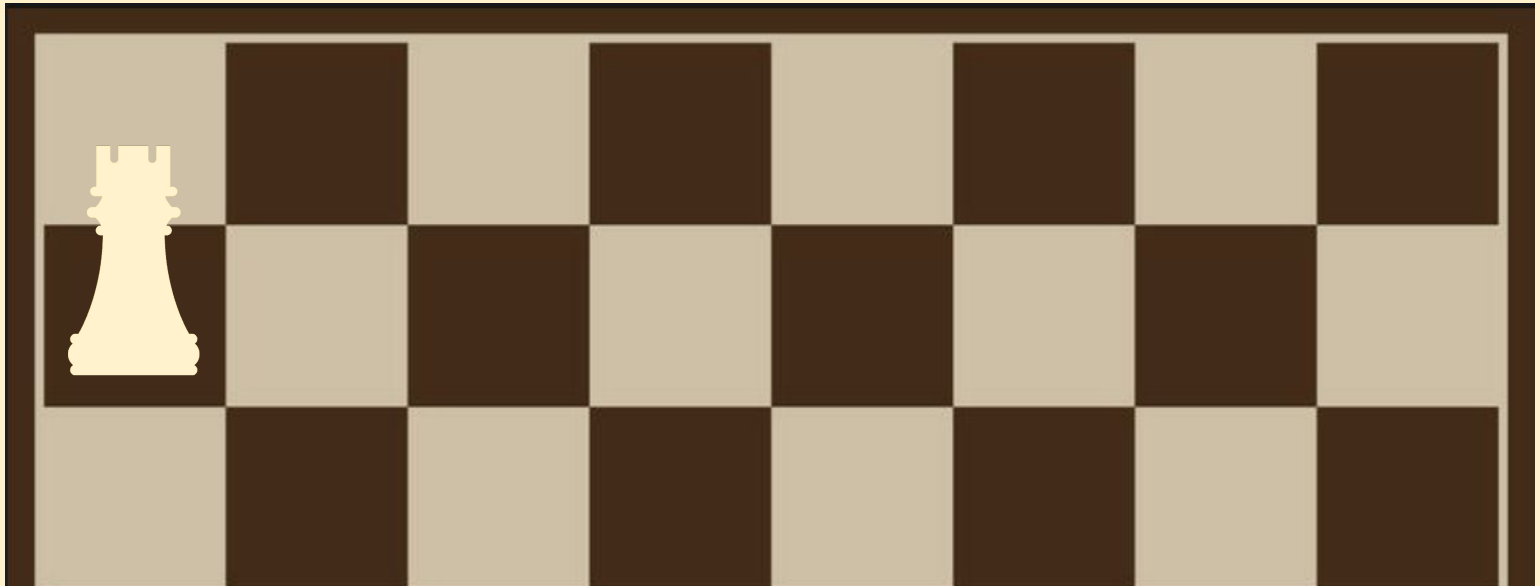
Oscar Bedford - 260792223

Saif Shahin - 260964749



Introduction & Problem definition

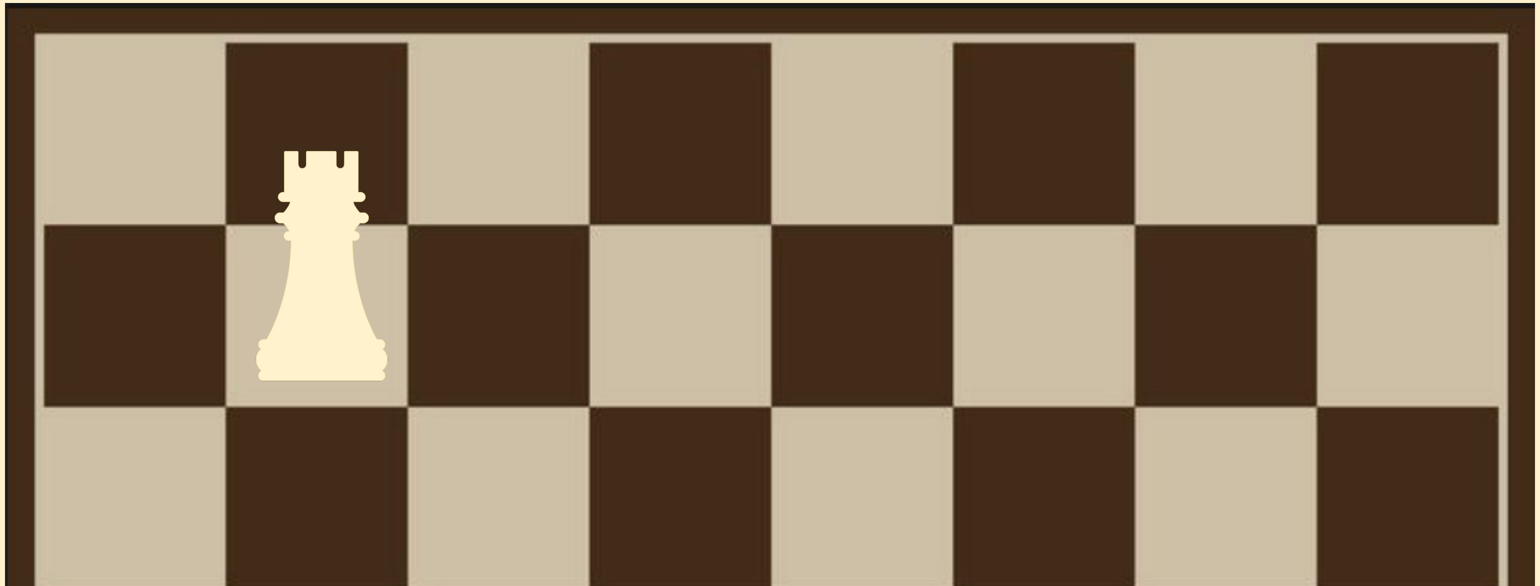
Chess annotation is important



Introduction & Problem definition

Chess annotation is important

Automation is difficult

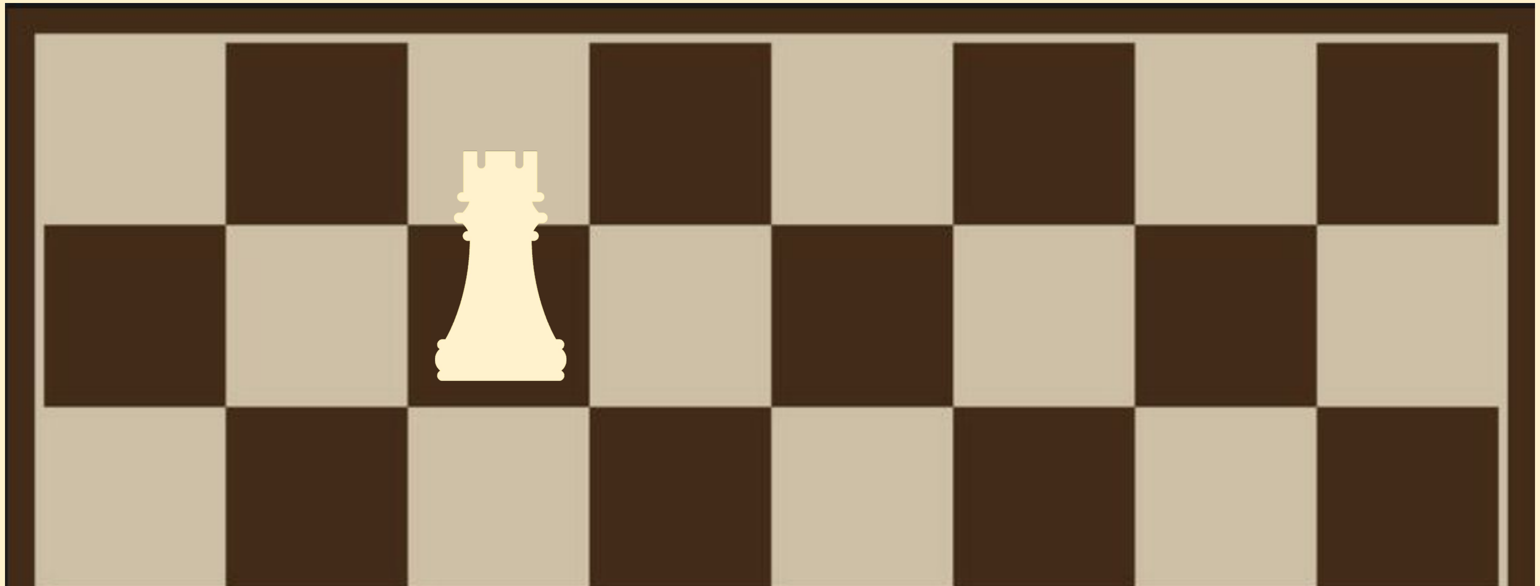


Introduction & Problem definition

Chess annotation is important

Automation is difficult

Requires a lot of labeled data



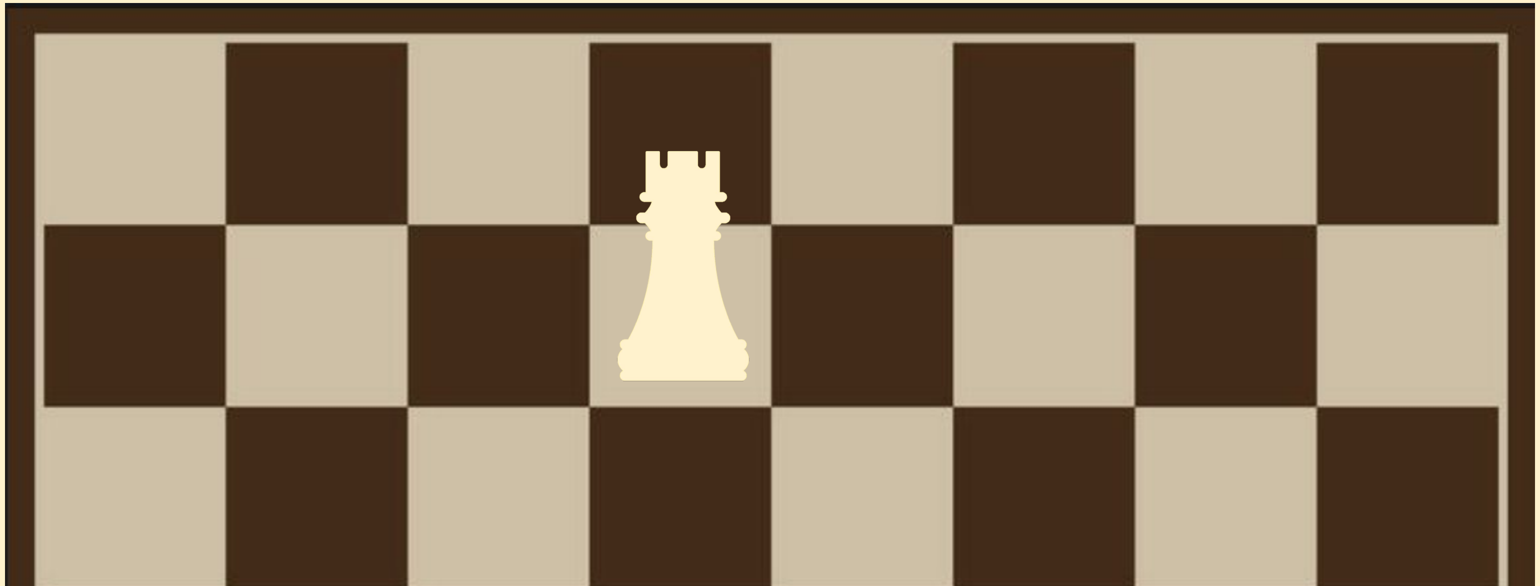
Introduction & Problem definition

Chess annotation is important

Automation is difficult

Requires a lot of labeled data

Labeled data is <u>hard</u> to find



Introduction & Problem definition

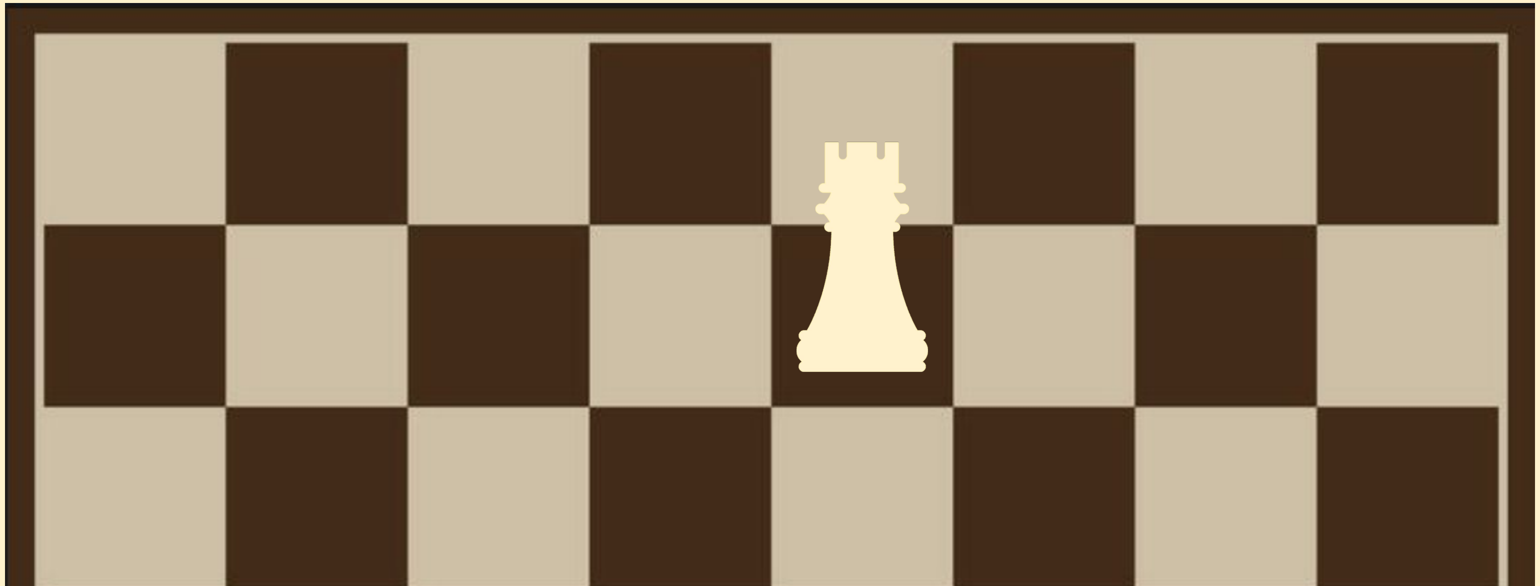
Chess annotation is important

Automation is difficult

Requires a lot of labeled data

Labeled data is hard to find

Labeling data is tedious



Introduction & Problem definition

Chess annotation is important

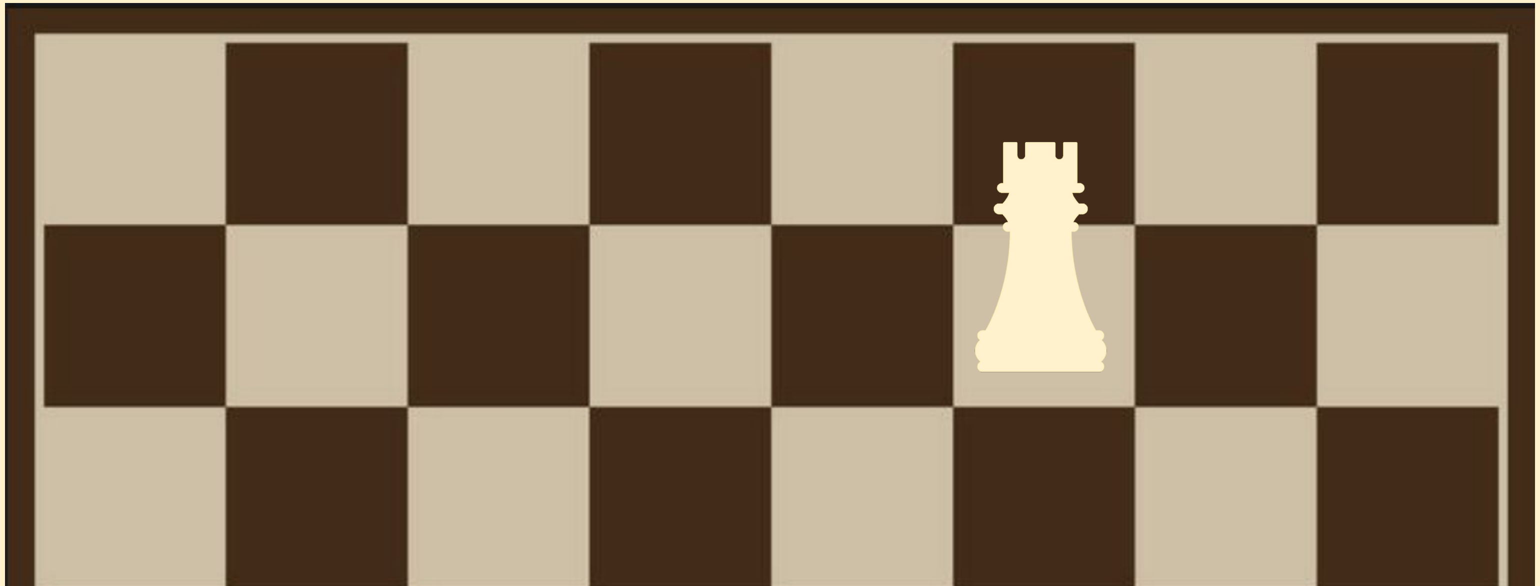
Automation is difficult

Requires a lot of labeled data

Labeled data is hard to find

Labeling data is tedious

Unsupervised domain adaptation



Overview of unsupervised DA[1]

- Domains
  - Target domain
    - Unlabeled, scarce
    - Data we want to predict
  - Source domain
    - Labeled, abundant
    - Helps to predict in the target domain

[1] S. Ben-David et al., "A theory of learning from different domains," Mach. Learn., vol. 79, pp. 151-175, 2010.

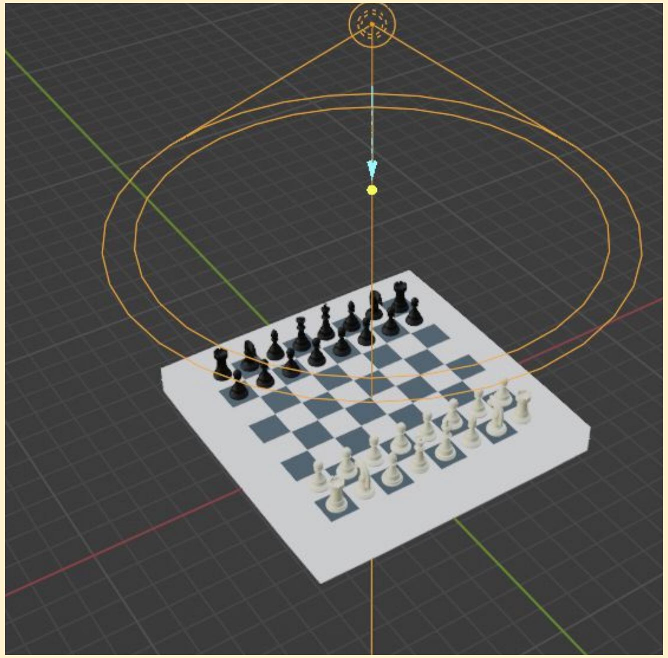
Goals & constraints

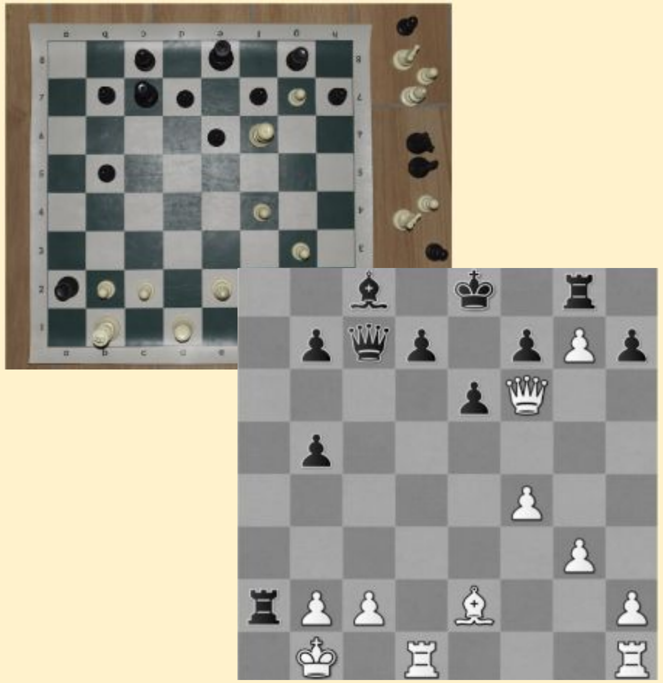
Generate valid 3D data → Build end-to-end pipeline → Benchmark DA performance

Image content Image format

Unprocessed target data

No DA model DA model 2





Classification accuracy

Weighted F1 score

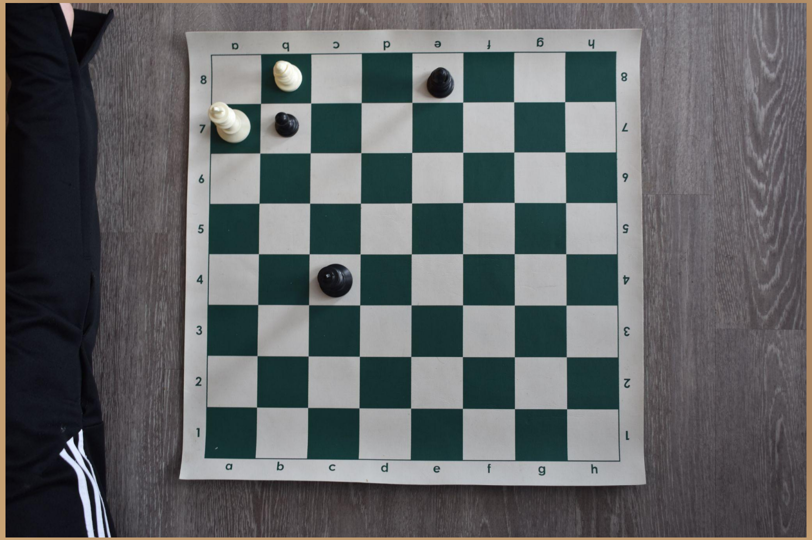
Confusion matrices

Methodology: Datasets

Real-life dataset[2]

Target domain : Domain where the model must perform well but lacks labeled data.

Labels were provided but were only used for hyperparameter tuning and testing in order to simulate an unsupervised domain adaptation situation.



500 training examples

Changes in Lighting

Changes in Background

[2] A. D. S. D. Neto and R. M. Campello, "Chess position identification using pieces classification based on synthetic images generation and deep neural network fine-tuning," in Proc. 21st Symp. Virtual Augmented Reality (SVR), pp. 152-160, IEEE, Oct. 2019.

Methodology: Datasets

Generated dataset

Source domain: 288,000 generated images

Various lighting conditions

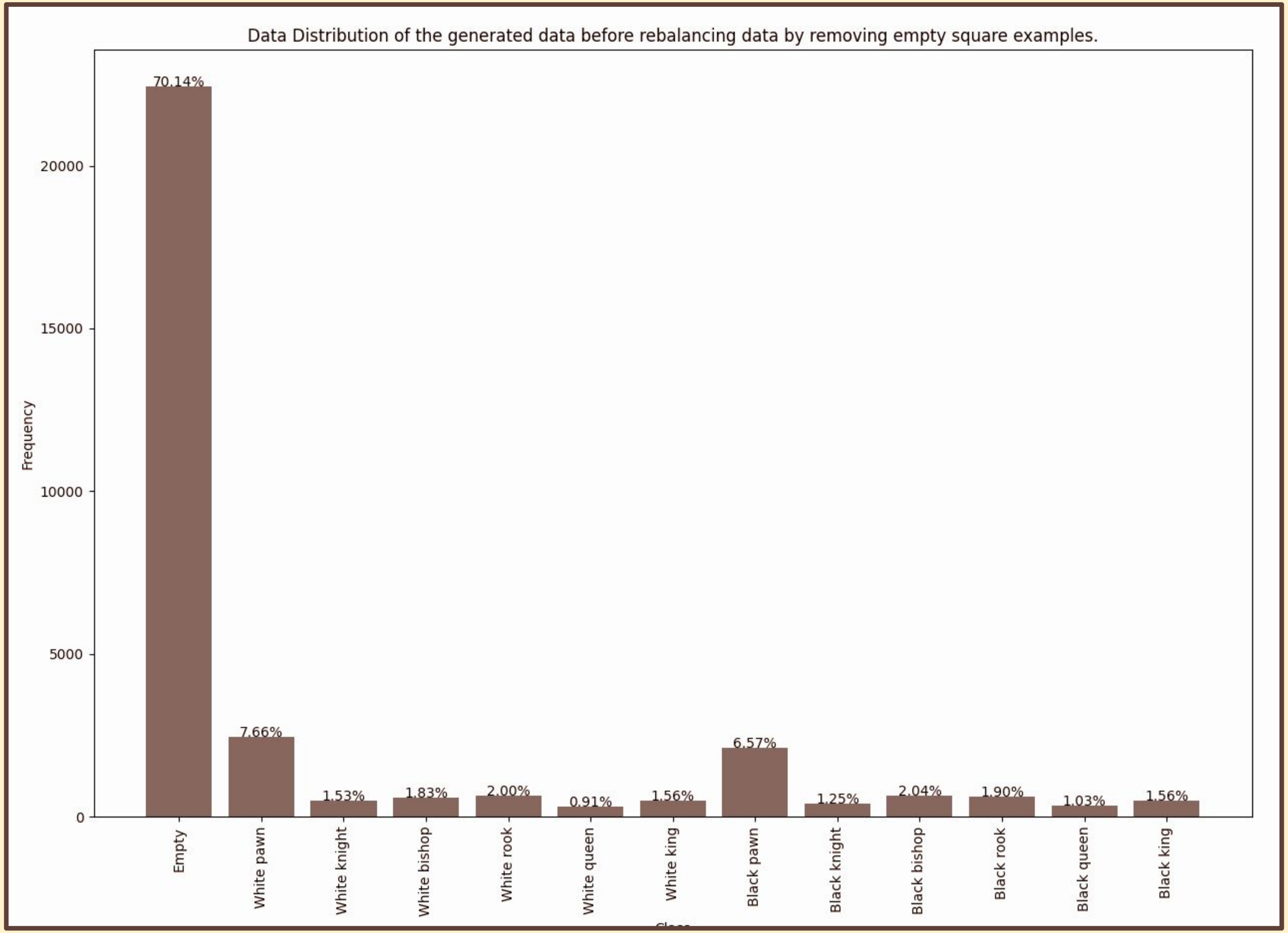
Different centering piece positions

Diverse piece rotations

Multiple camera angles

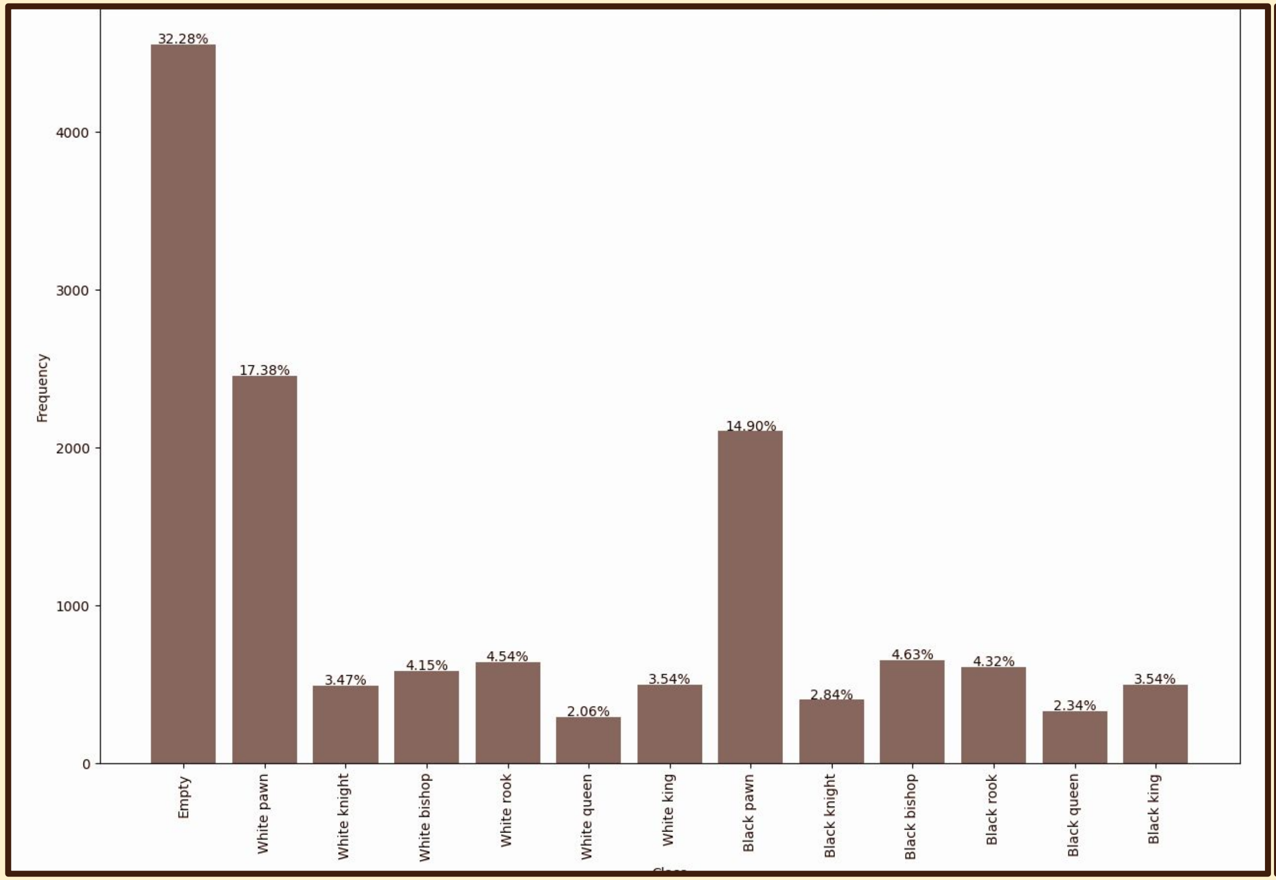
Makes the trained model <u>invariant</u> to all of the above conditions

Methodology: Datasets

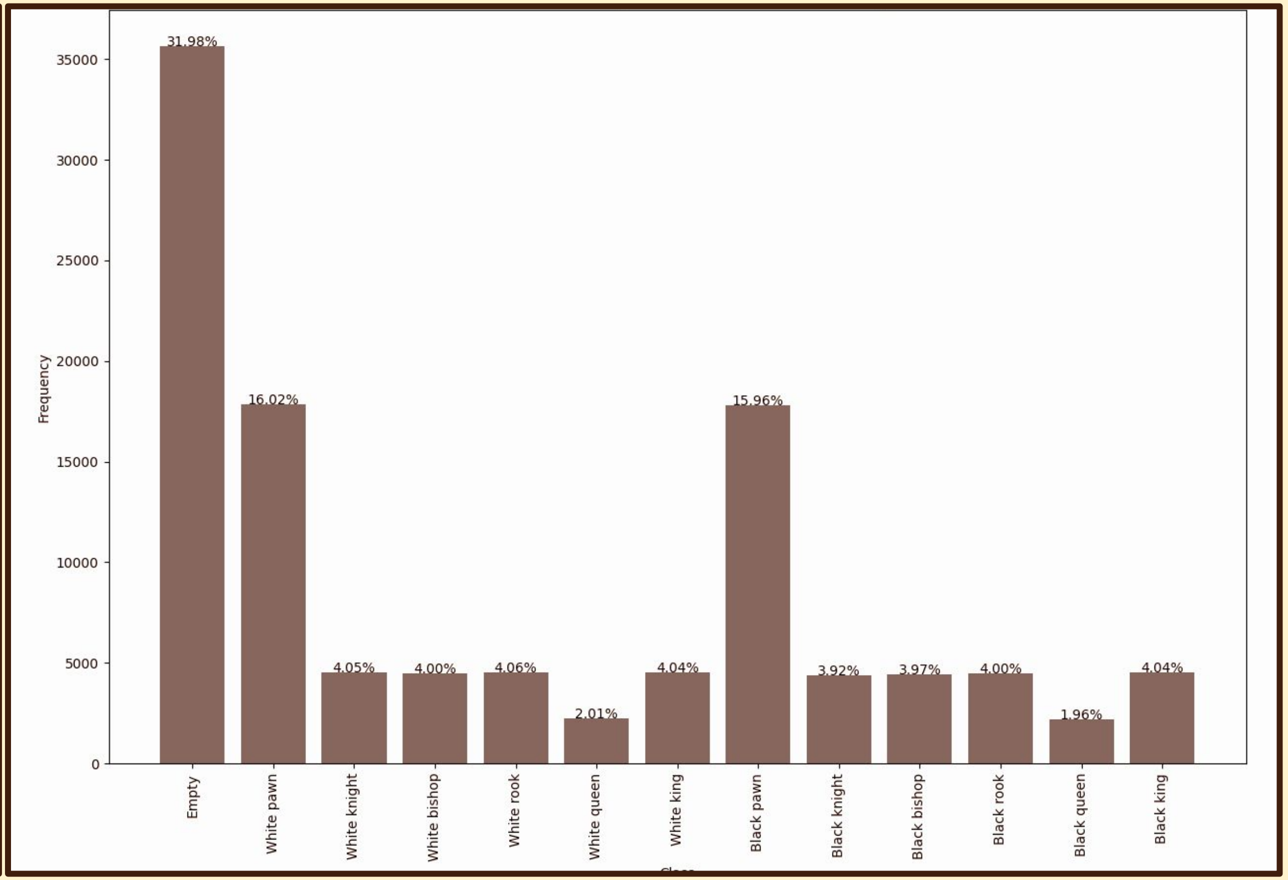


Methodology: Datasets

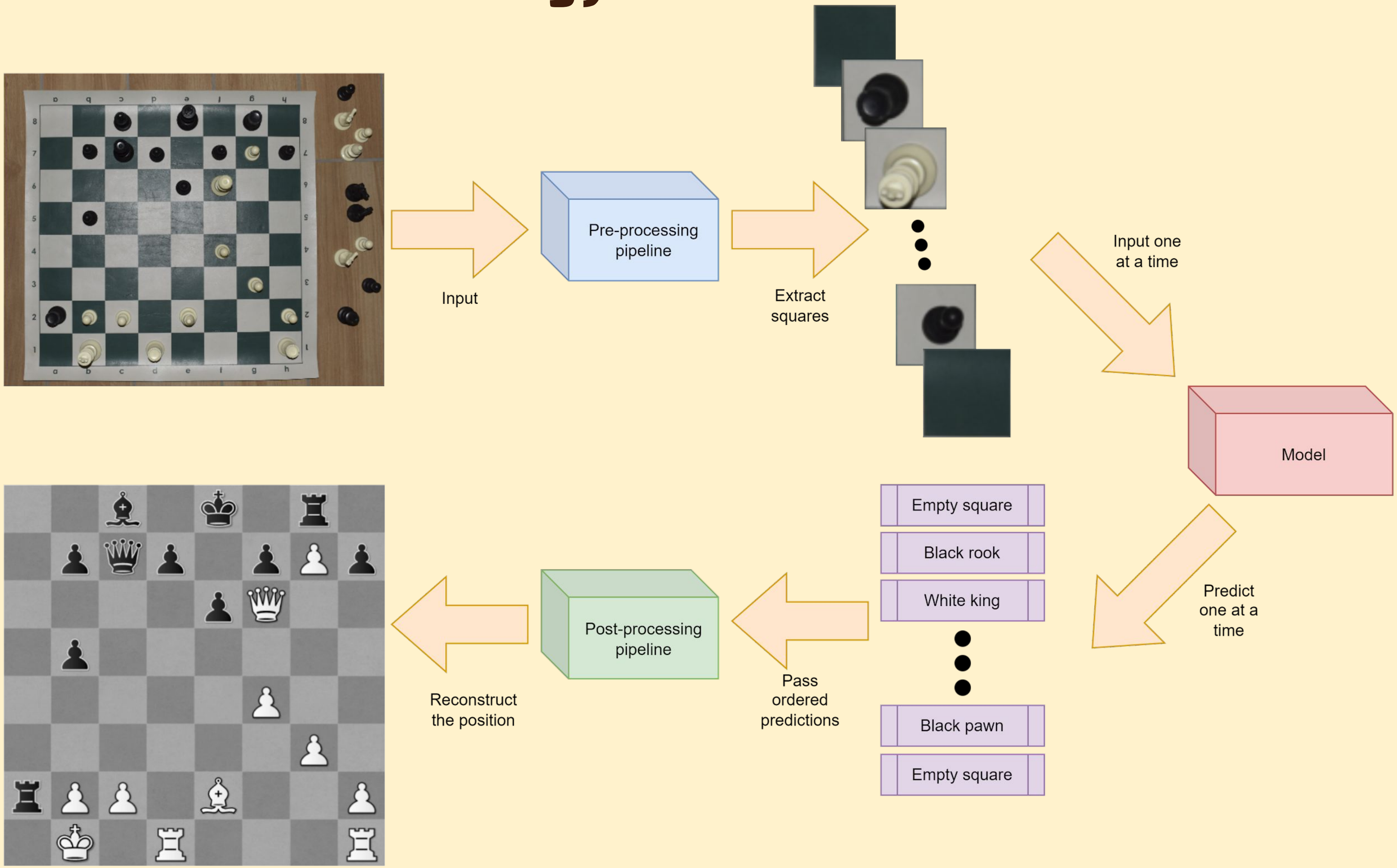
Data distribution of the real life dataset after rebalancing



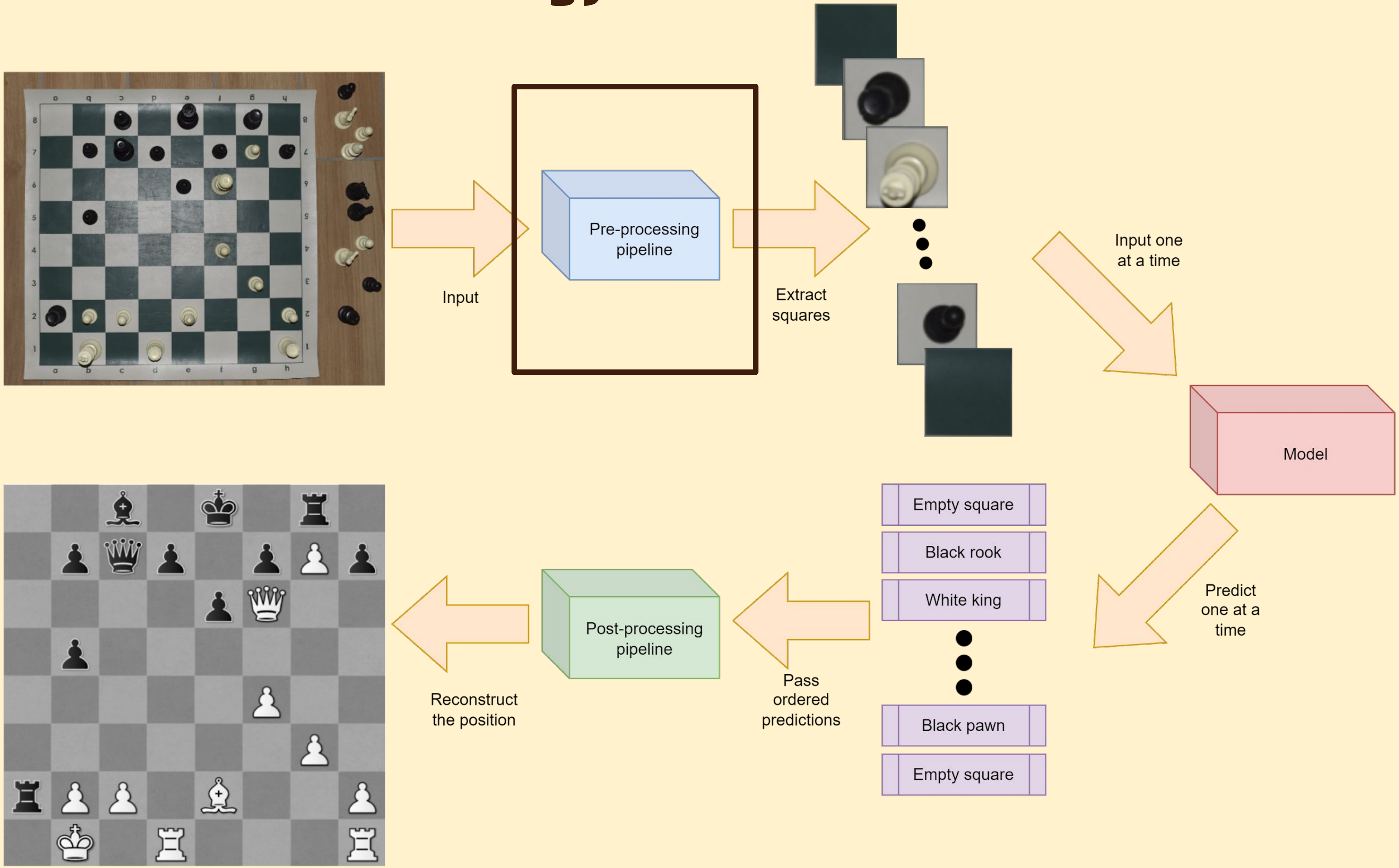
Data distribution of the generated dataset



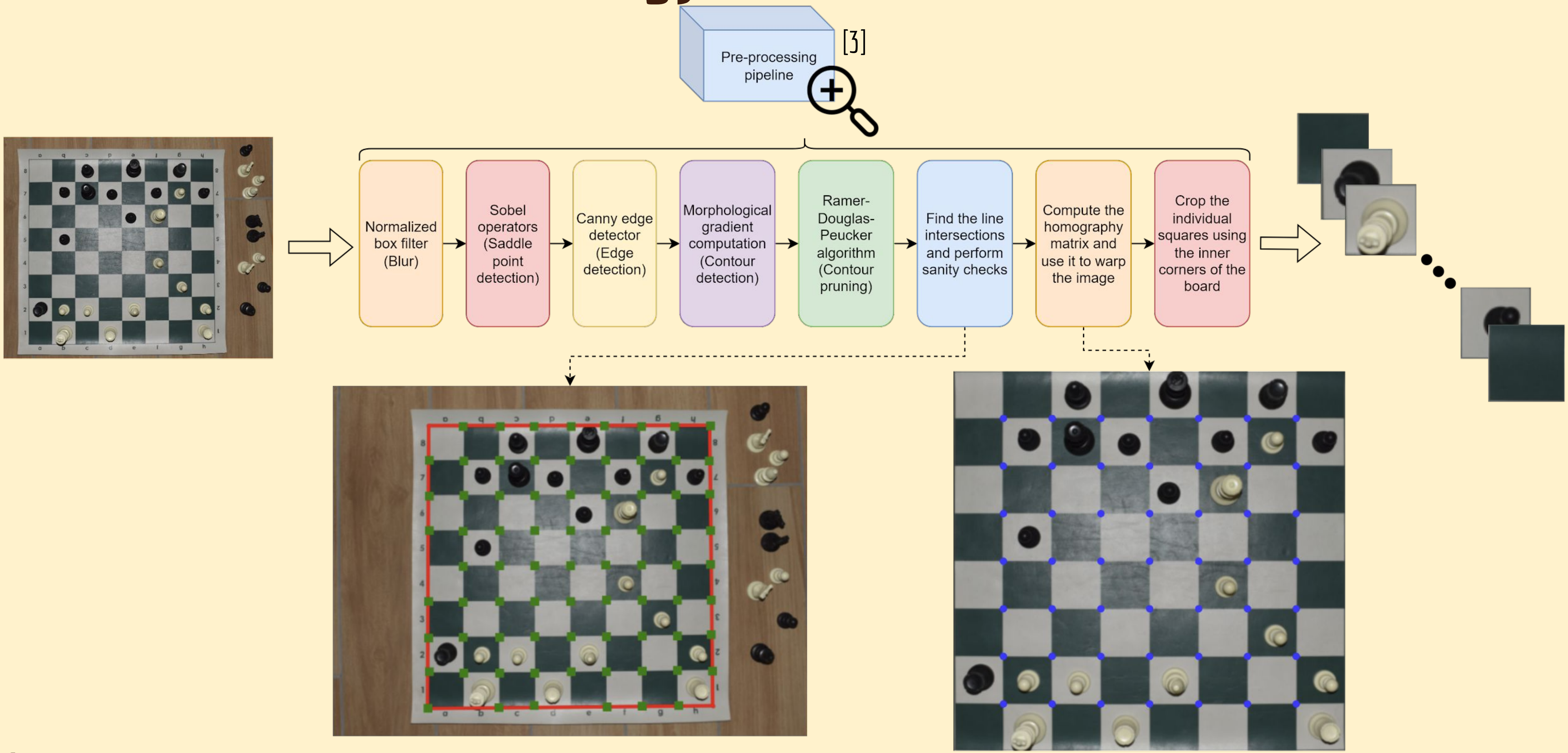
Methodology: The architectures



Methodology: The architectures

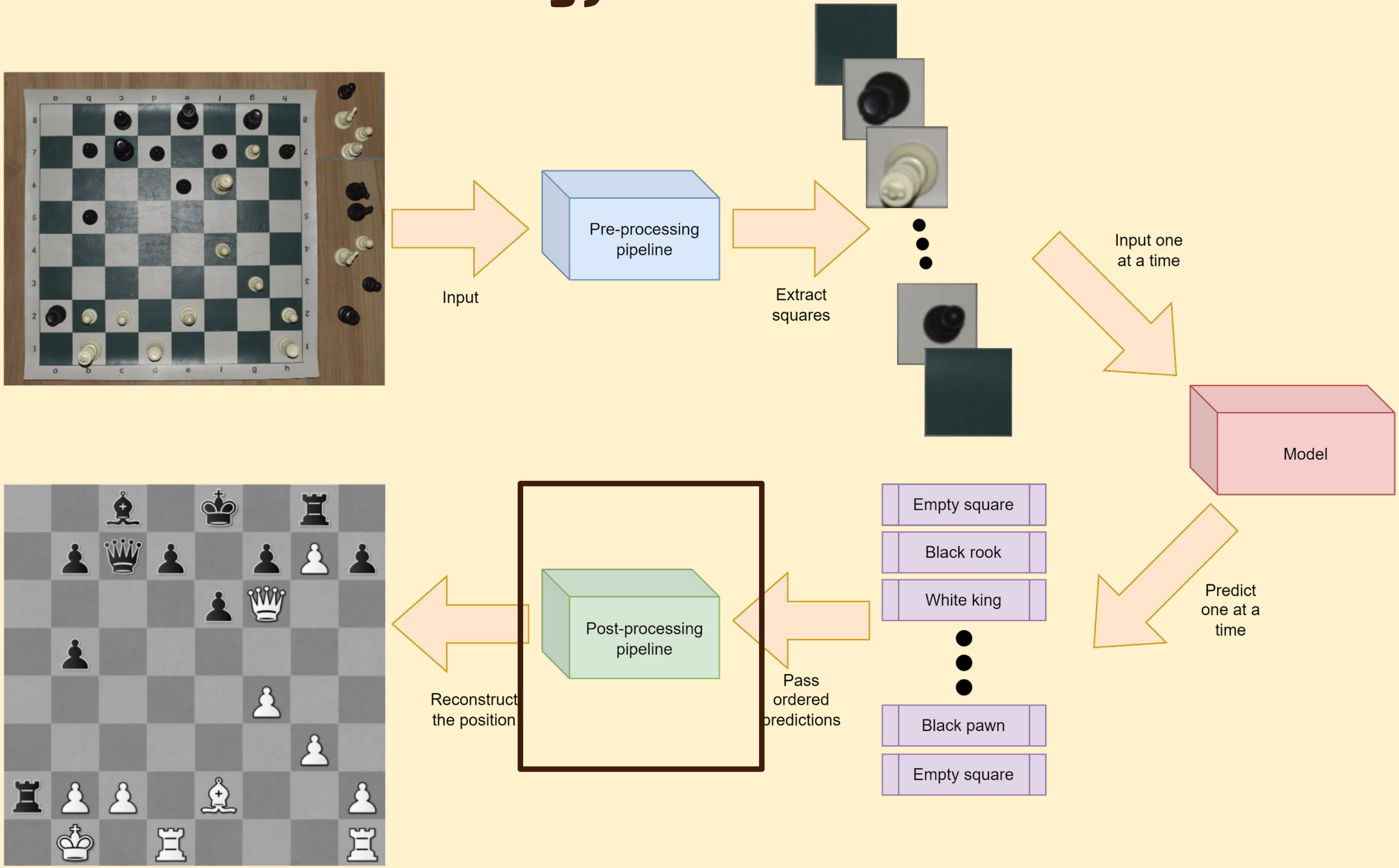


Methodology: The architectures

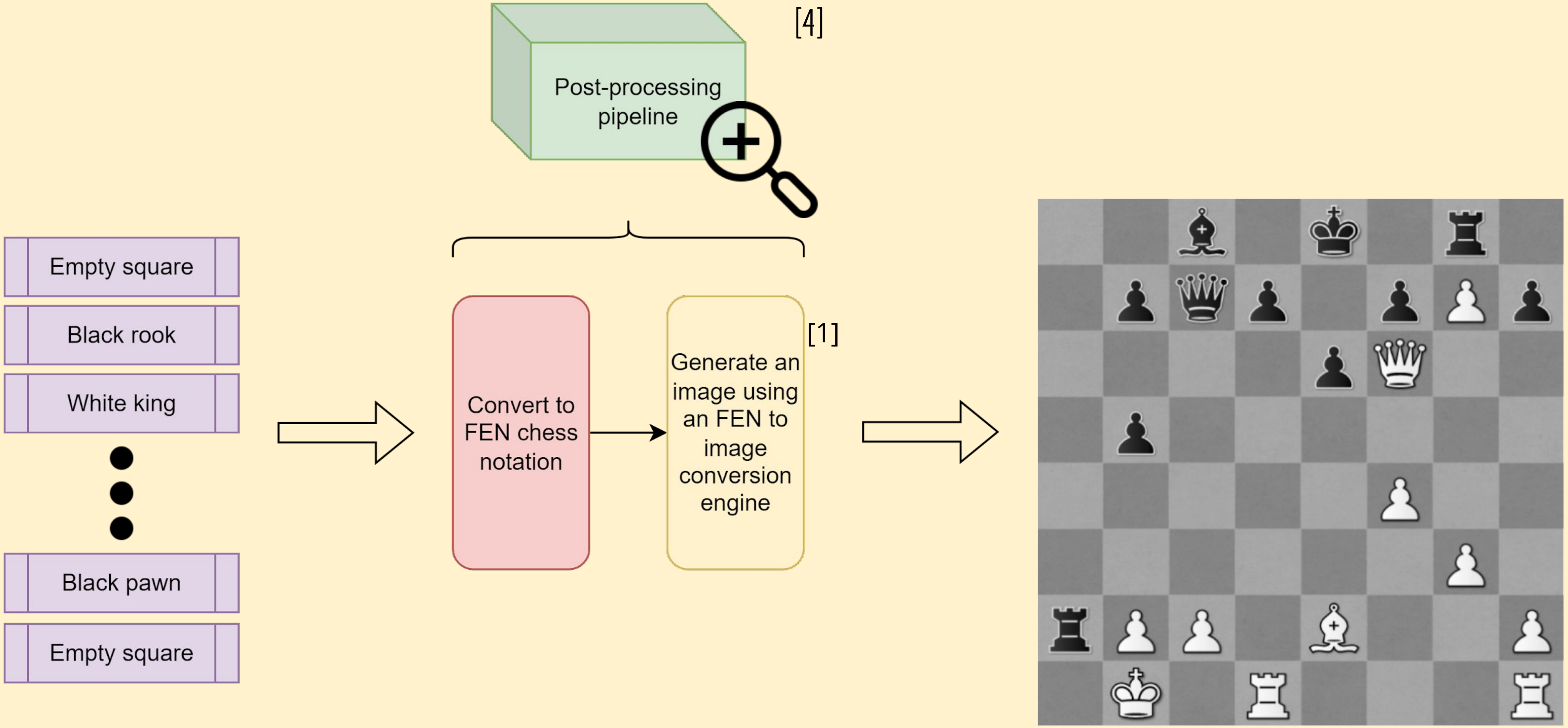


[3]: Elucidation, "Elucidation/chessboarddetect: Hodgepodge of chessboard detection algorithms on images from actual matches.," *GitHub*. [Online]. Available: https://github.com/Elucidation/ChessboardDetect. [Accessed: 10-Apr-2023].

Methodology: The architectures

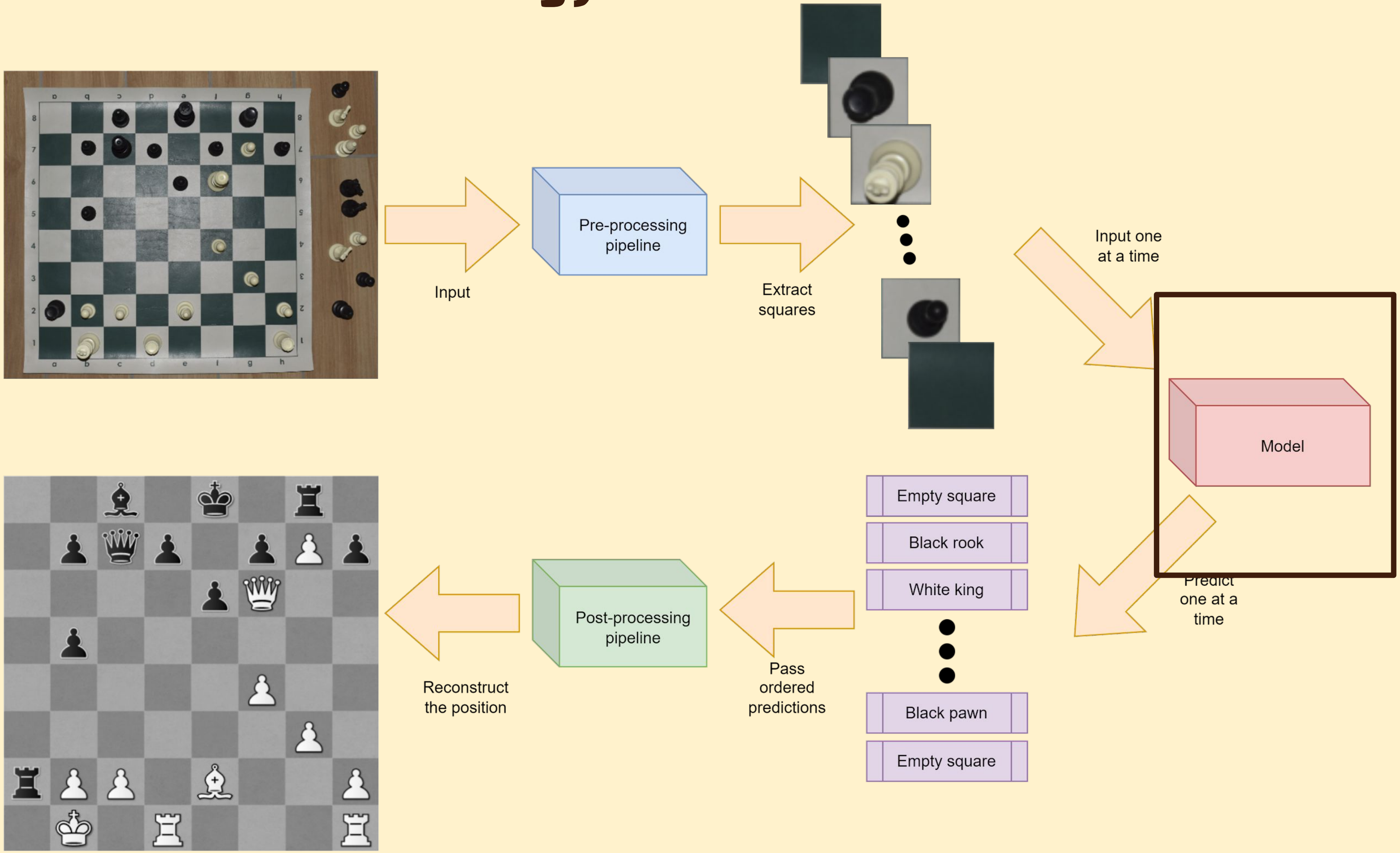


Methodology: The architectures



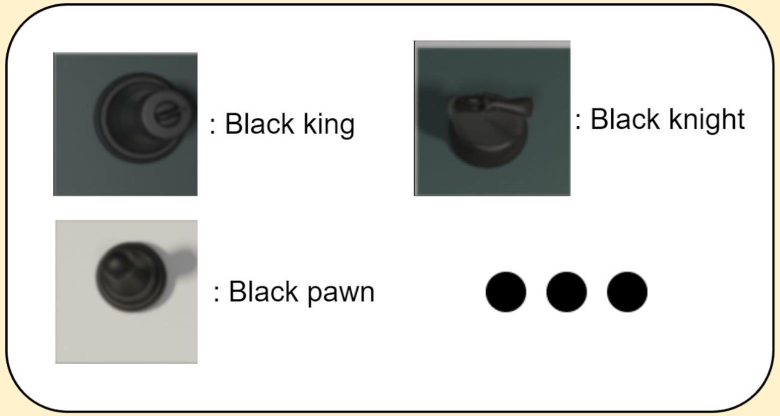
[4] "The best free, adless chess server," *lichess.org*. [Online]. Available: https://lichess.org/. [Accessed: 10-Apr-2023].

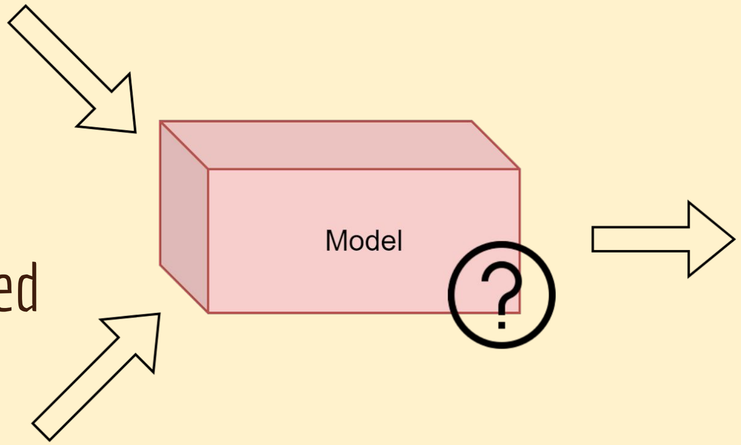
Methodology: The architectures



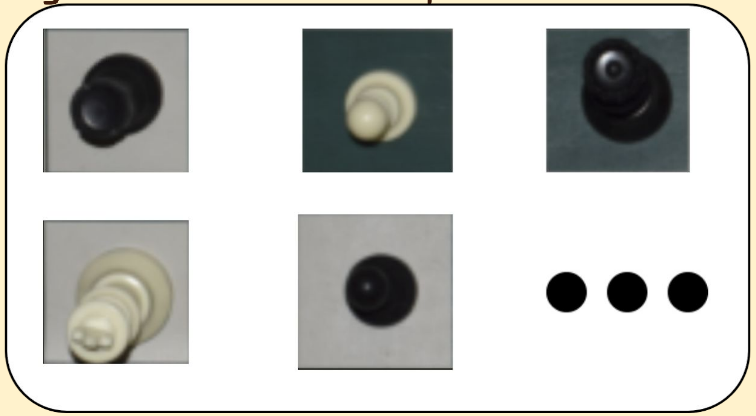
Methodology: The architectures

Source domain: Generated data, labelled





Target domain: Real life data, unlabelled

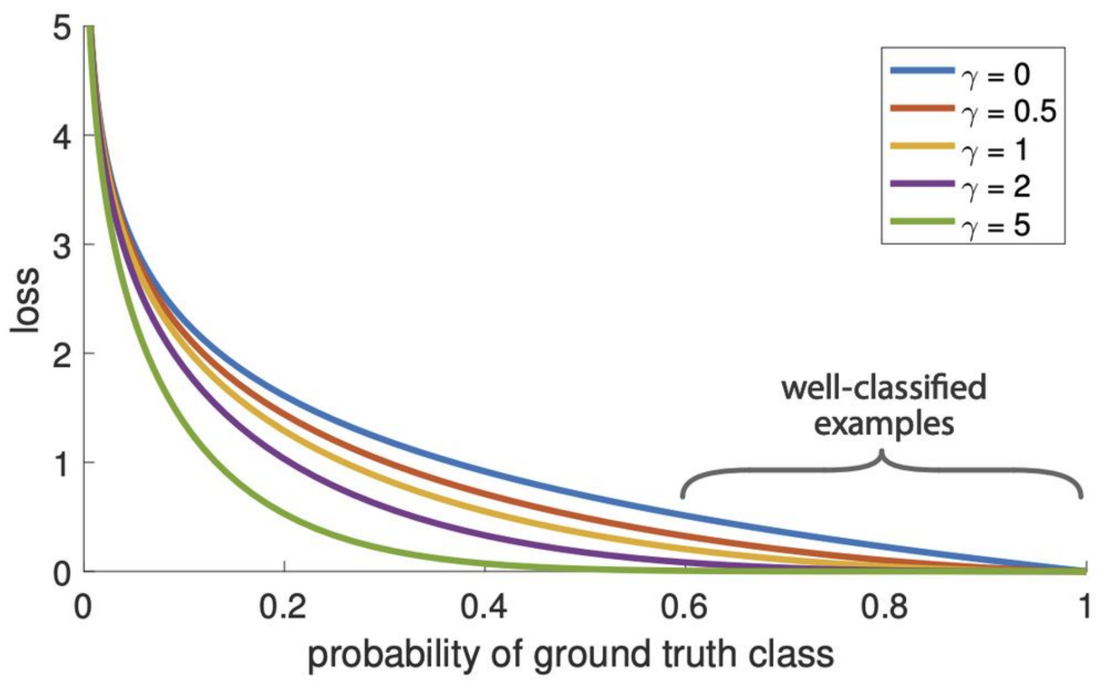


What domain adaptation technique to use in order to get good accuracies on the target domain?

Methodology: The loss functions

Cross Entropy = $-\ log(p_i)$

Focal Loss [5] = $-\ (1-p_i)^{\gamma} log(p_i)$



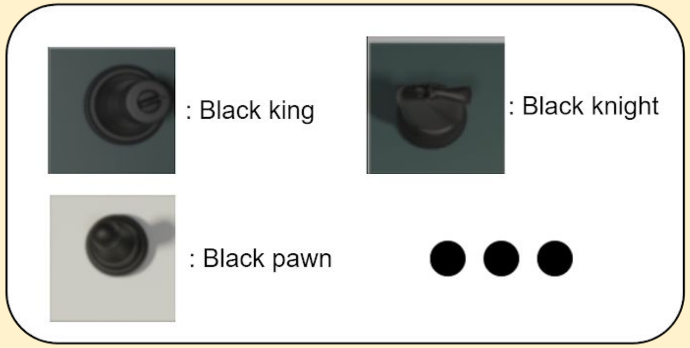
CORAL Loss [6] = $\|C_s - C_t\|_F^2$ with C(X) = $\frac{1}{n-1}(X-\overline{X})^T(X-\overline{X})$

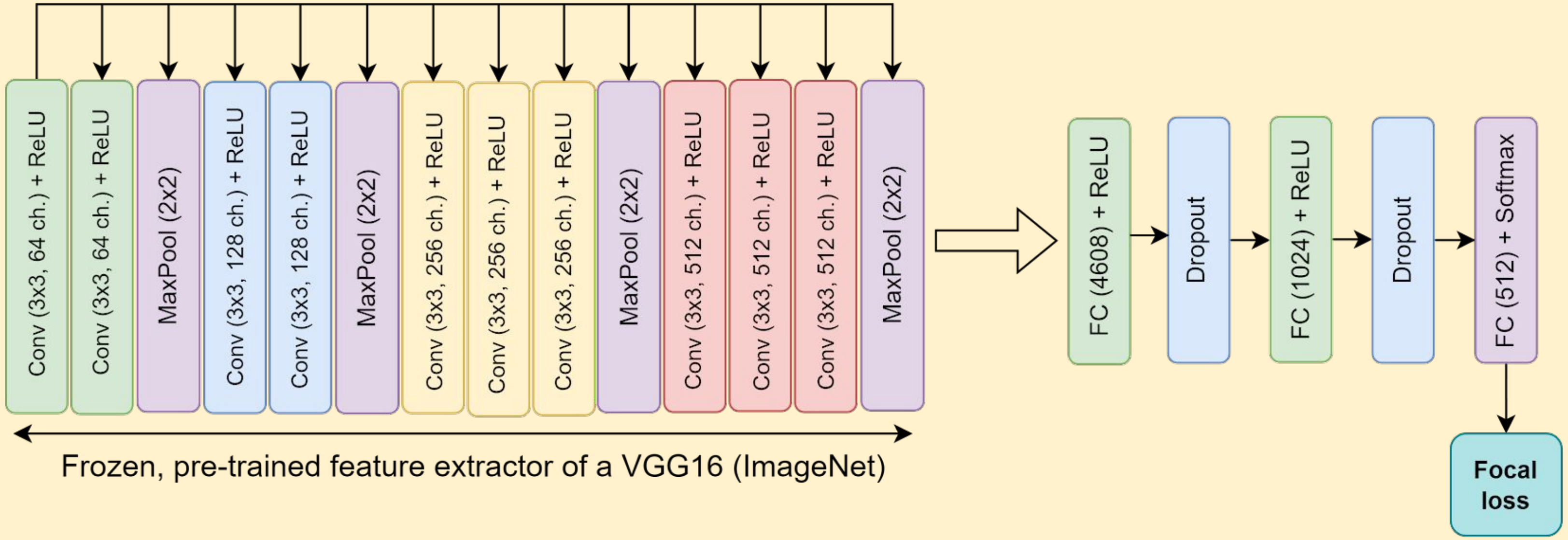
[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, 'Focal Loss for Dense Object Detection', *arXiv [cs.CV]*. 2018.
[6] B. Sun, J. Feng, and K. Saenko, 'Correlation Alignment for Unsupervised Domain Adaptation', *CoRR*, vol. abs/1612.01939, 2016.

Methodology: The architectures

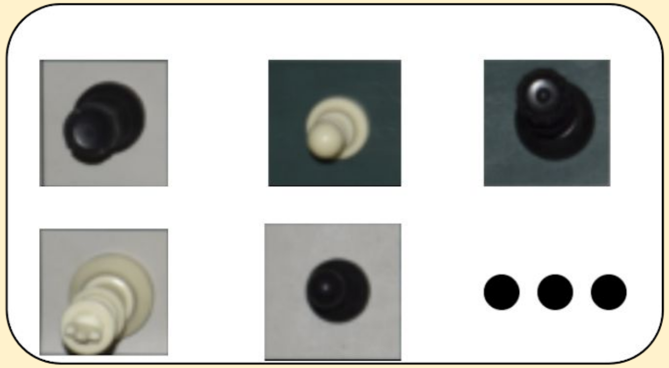
Baseline 1: VGG16[7] without domain adaptation

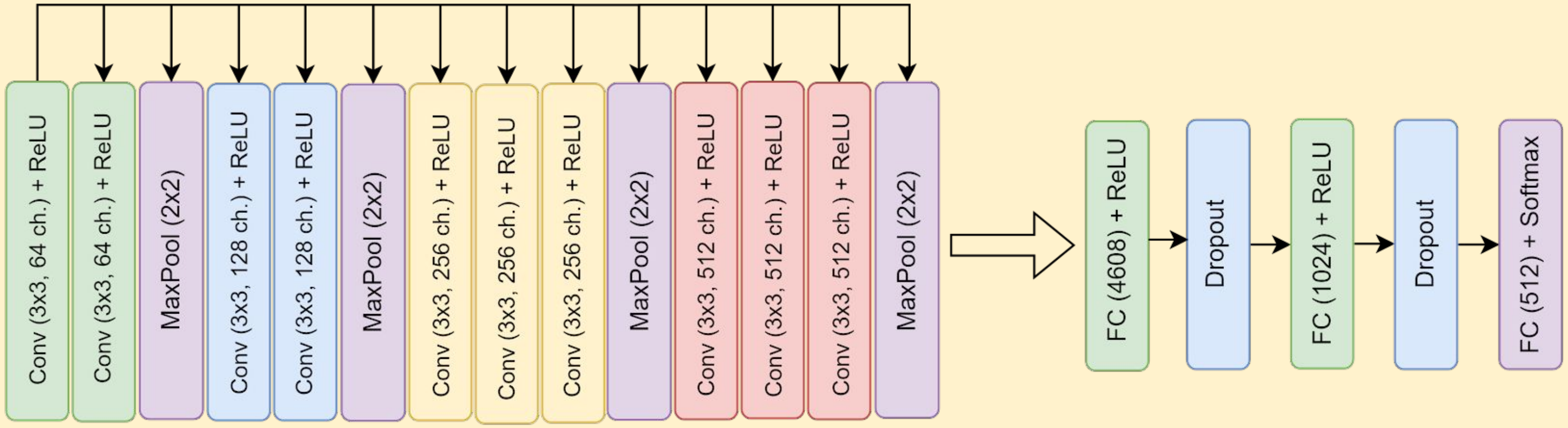
During training





During testing





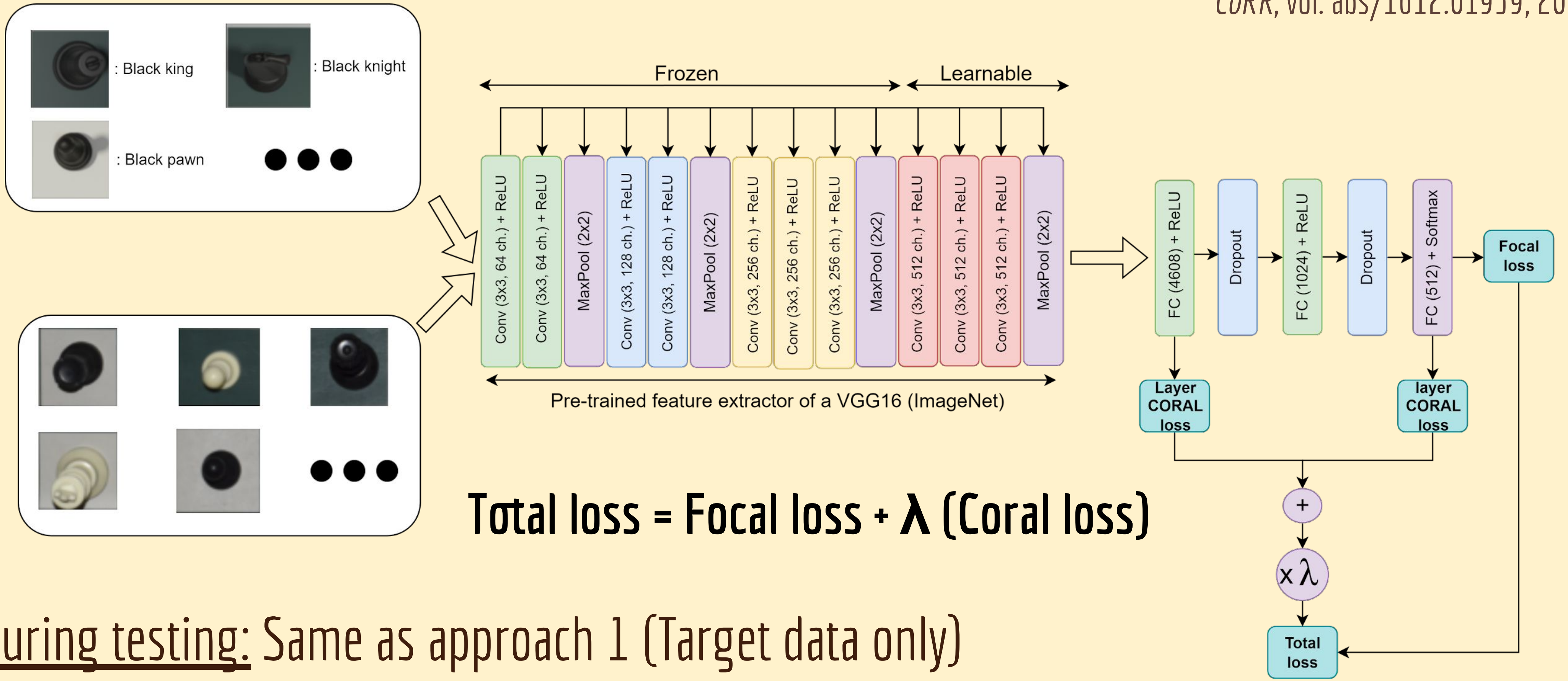
[7] K. Simonyan and A. Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition', CoRR, vol. abs/1409.1556, 2014.

Methodology: The architectures

Baseline 2: Correlation alignment (CORAL) [8]

[8] B. Sun, J. Feng, and K. Saenko, 'Correlation Alignment for Unsupervised Domain Adaptation', *CoRR*, vol. abs/1612.01939, 2016.

<u>During training</u>



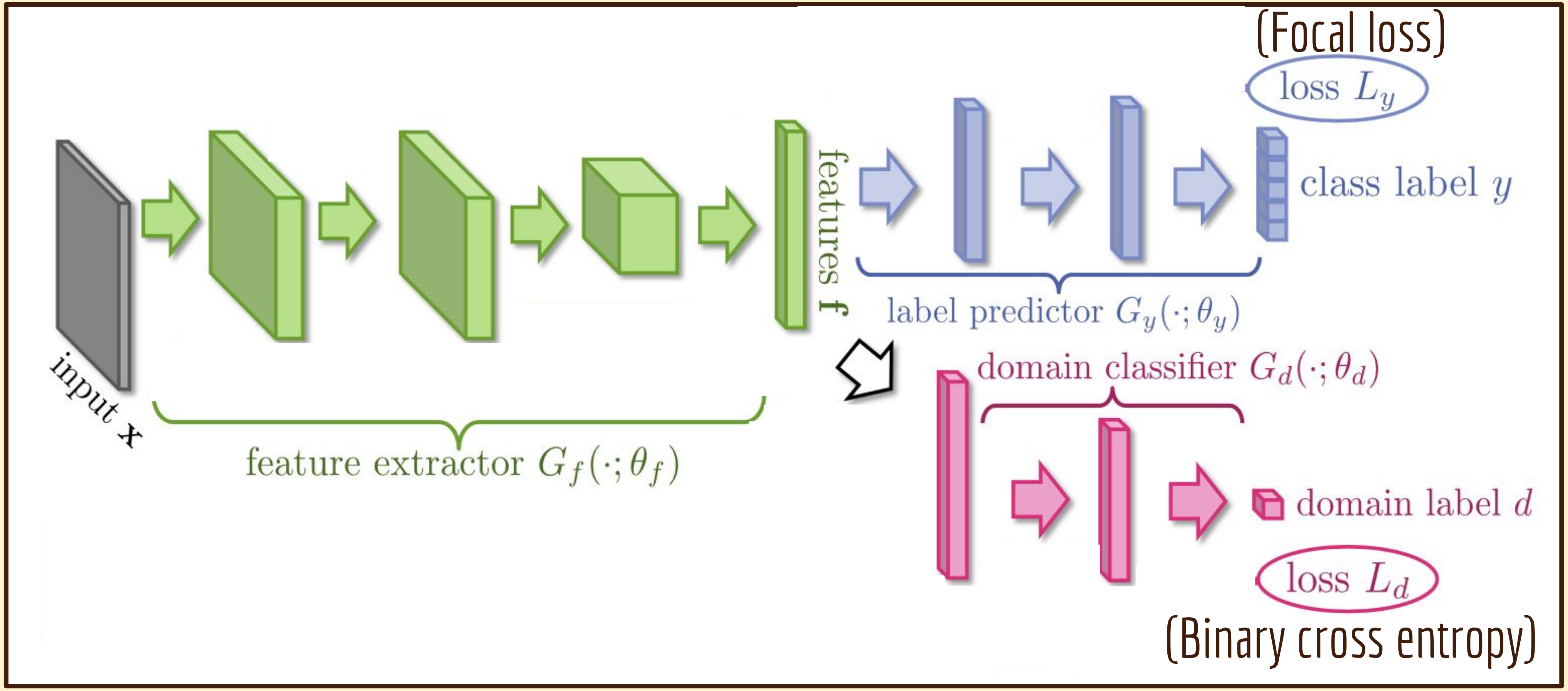
Total loss = Focal loss + λ (Coral loss)

<u>During testing:</u> Same as approach 1 (Target data only)

Methodology: The architectures

Our approach: Adversarial training (DANN) [9]

During training

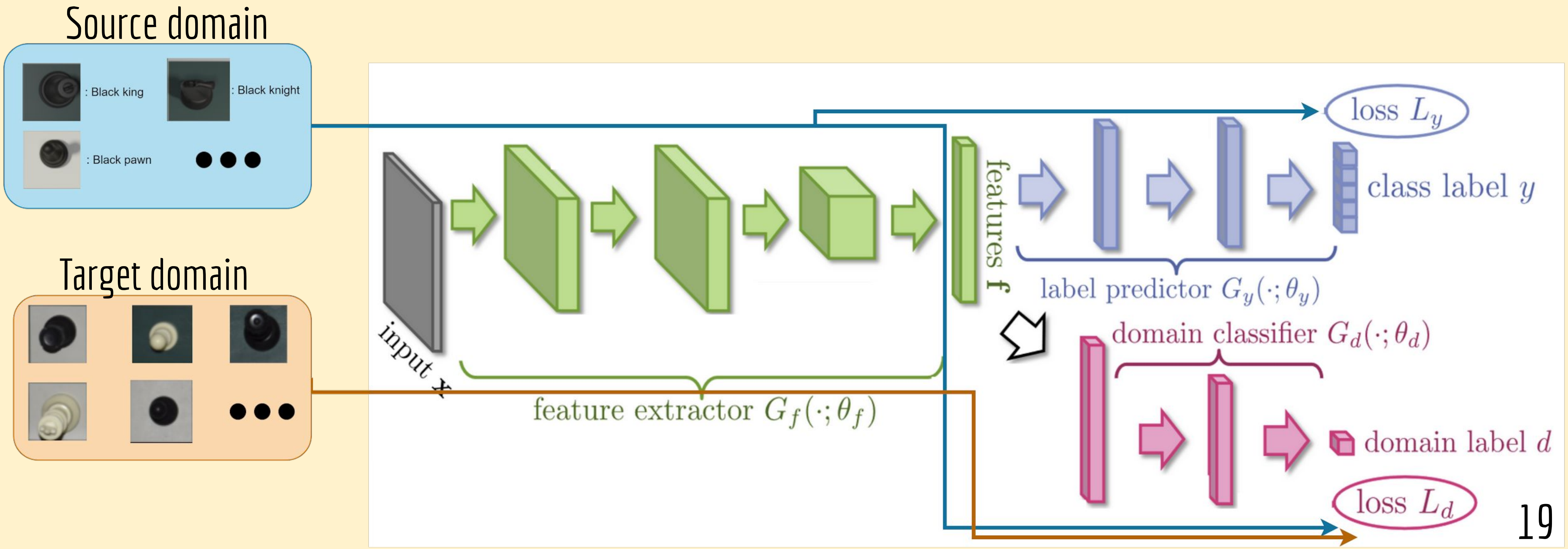


[9] Y. Ganin *et al.*, 'Domain-Adversarial Training of Neural Networks', *arXiv [stat.ML]*. 2016.

Methodology: The architectures

Our approach: Adversarial training (DANN)

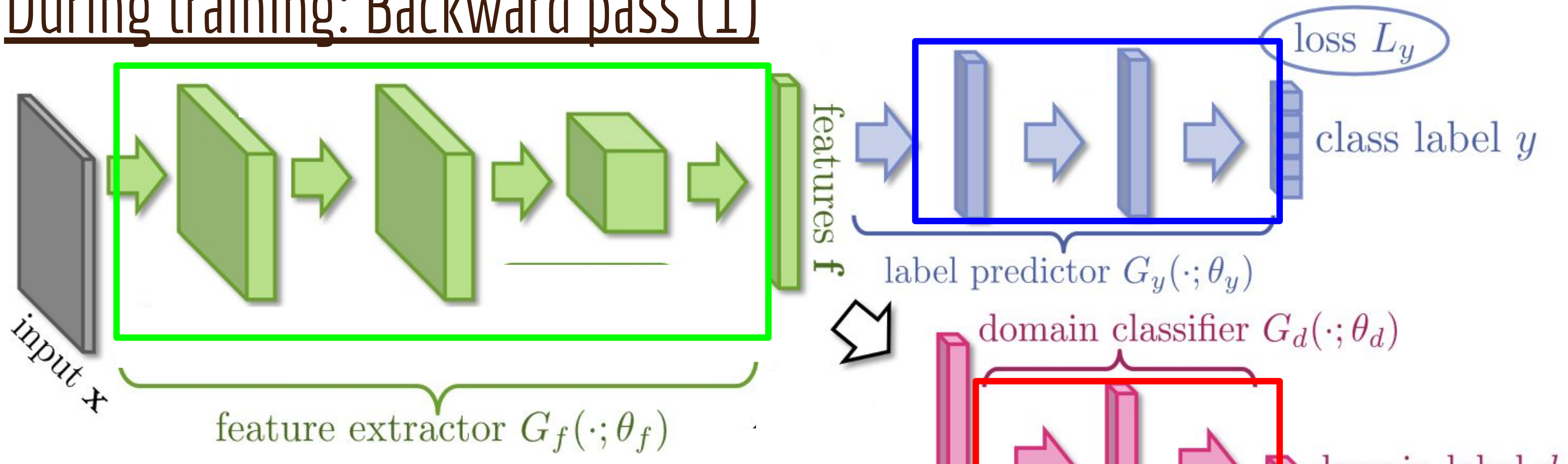
During training: Forward pass



Methodology: The architectures

Our approach: Adversarial training (DANN)

During training: Backward pass (1)



Optimizes $L_f = L_y \boxed{-} \lambda L_d$

$$\theta_f = \theta_f - \frac{\partial L_y}{\partial \theta_f} \boxed{+} \lambda \frac{\partial L_d}{\partial \theta_f}$$

Optimizes L_y

$$\theta_y = \theta_y - \frac{\partial L_y}{\partial \theta_y}$$

Optimizes L_d

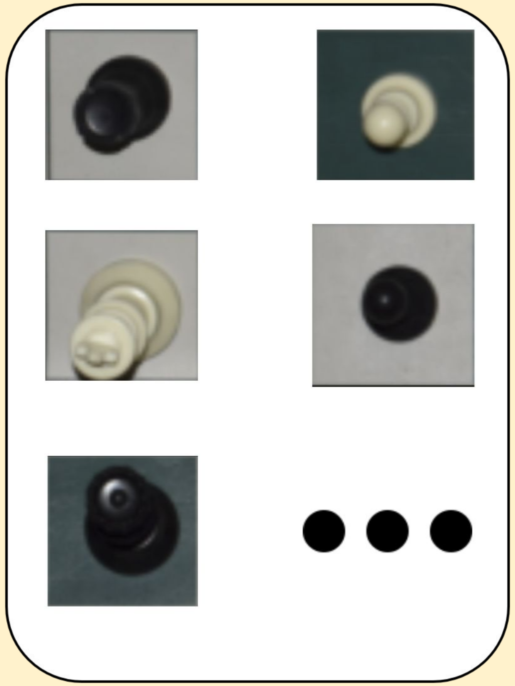
$$\theta_d = \theta_d - \frac{\partial L_d}{\partial \theta_d}$$

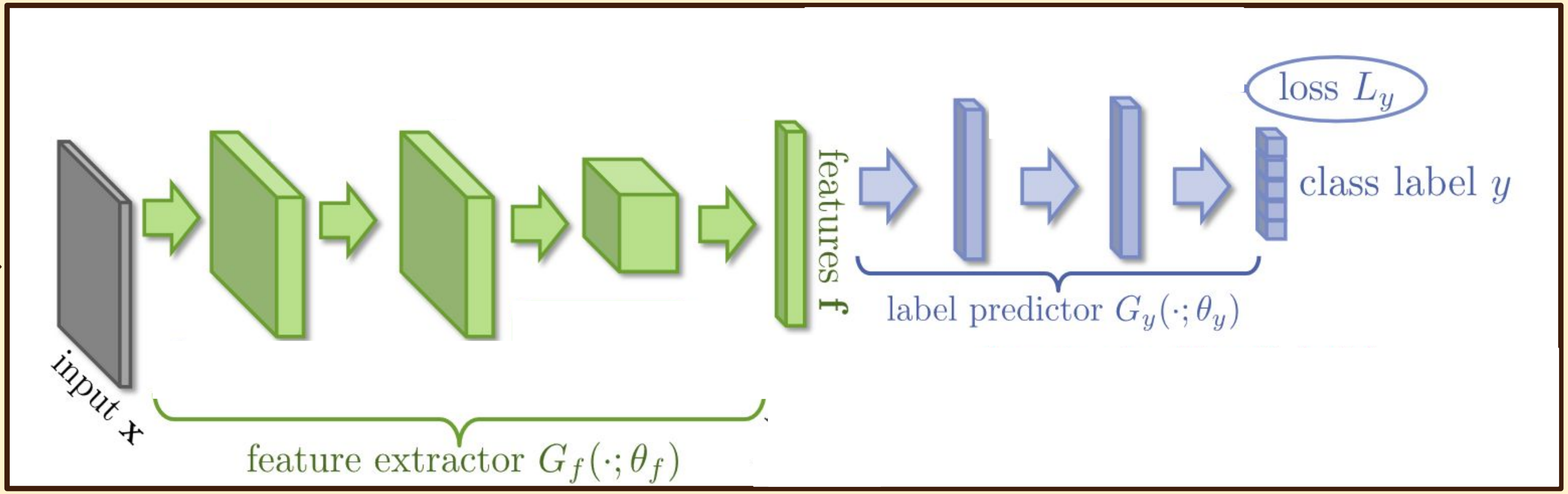
Methodology: The architectures

Our approach: Adversarial training (DANN)

During testing

Target domain





Results: Hyperparameter tuning

Gamma	Batch size	Dropout rate	Final validation accuracy
2.0	200.0	0.2	52.82%
2.0	100.0	0.2	56.67%
2.0	200.0	0.5	49.26%
2.0	100.0	0.5	51.76%
5.0	200.0	0.2	52.81%
5.0	100.0	0.2	46.53%
5.0	200.0	0.5	51.2%
5.0	**100.0**	**0.5**	**56.81%**

Base Model

Learning rate	Lambda max coral	Final validation accuracy
0.001	0.01	75.04%
0.001	0.1	83.15%
0.001	1.0	81.73%
0.001	**10.0**	**85.25%**
0.001	100.0	71.69%
0.01	0.01	9.35%
0.01	0.1	59.97%
0.01	1.0	39.52%
0.01	10.0	27.27%
0.01	100.0	7.76%

CORAL Model

Lambda (Domain adaptation)	Learning Rate	Final validation accuracy
0.1	0.075	74.0%
0.1	0.01	75.9%
variablepaper	0.075	80.5%
variablepaper	0.01	82.8%
0.2	0.0075	80.0%
0.2	**0.01**	**84.1%**

DANN Model

Results: Testing metrics on the source domain

Final **metrics** on the **oversampled generated data**

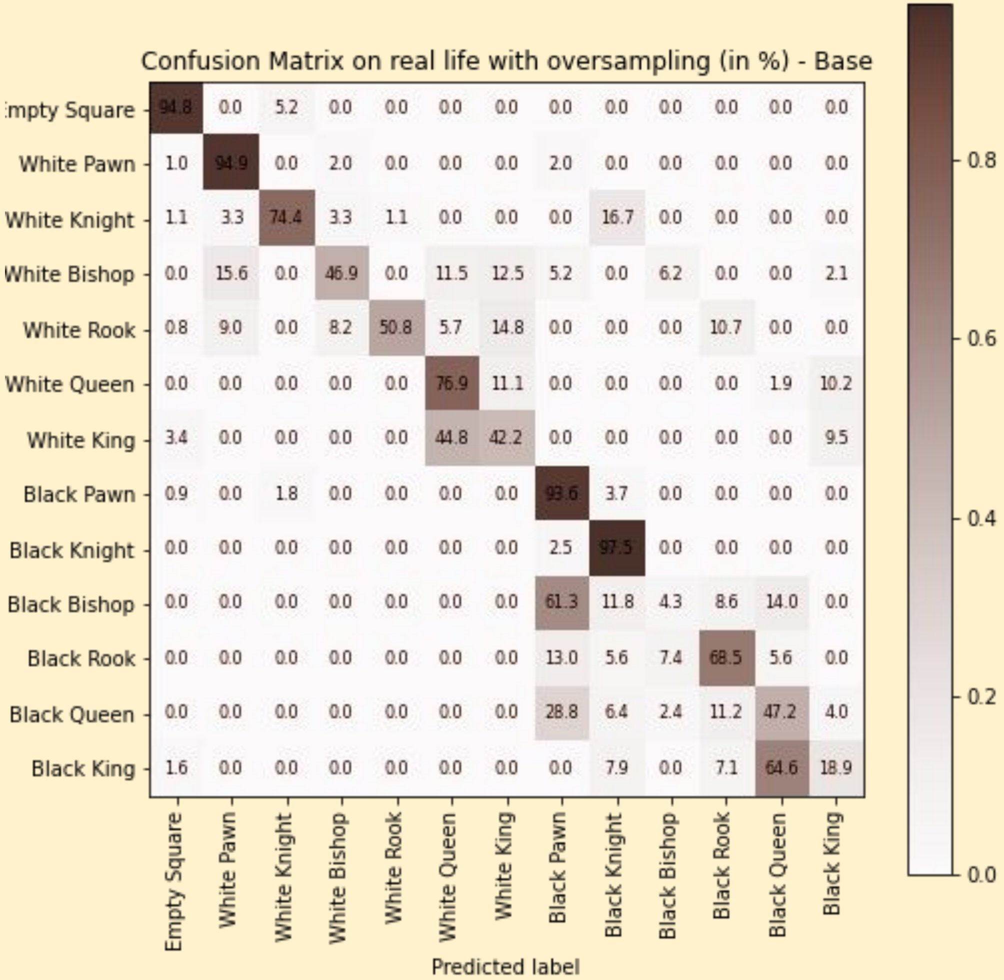
Models	Testing Accuracies	F1-Score	Average AUPRC
DANN	99.94%	0.999	0.999
CORAL	99.74%	0.997	0.997
Base Model	97.30%	0.972	0.974

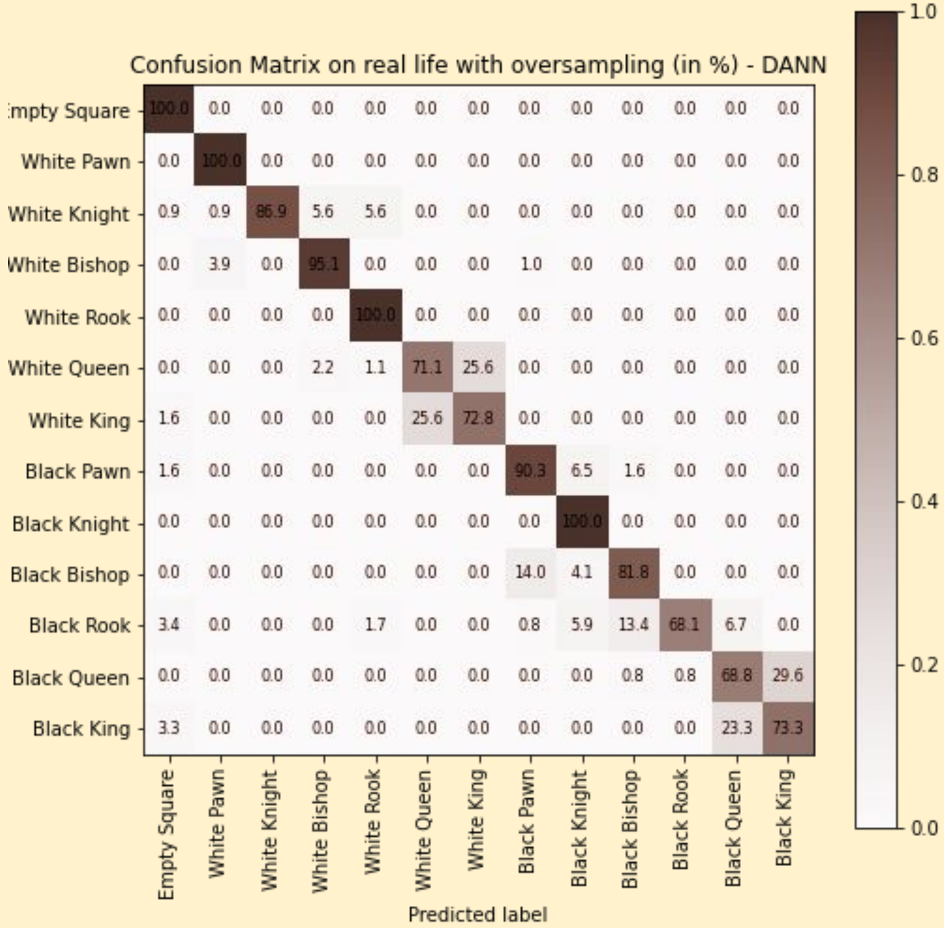
Results: Testing metrics on the target domain

Final **metrics** on the **oversampled real life data**

Models	Testing Accuracies	F1-Score	Average AUPRC
DANN	83.80%	0.847	0.858
CORAL	83.68%	0.835	0.847
Base Model	61.97%	0.592	0.641

Results: Confusion matrices with oversample





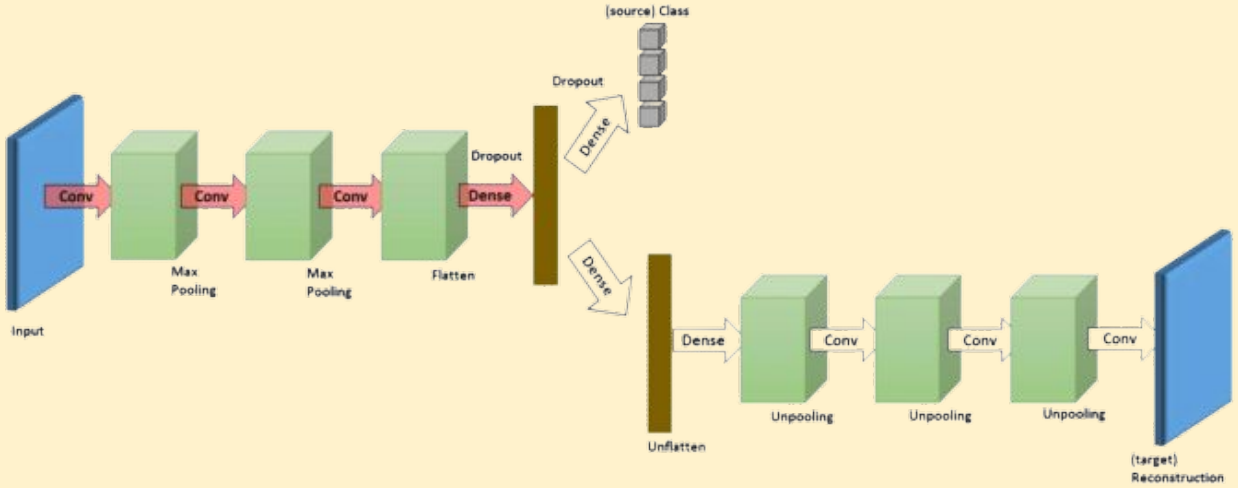
Conclusions

- First time domain adaptation is used in this application- and its great!
- Great results without having to label the training data
- Invariant to rotation, translation, camera angle, and lighting
- A LOT of data

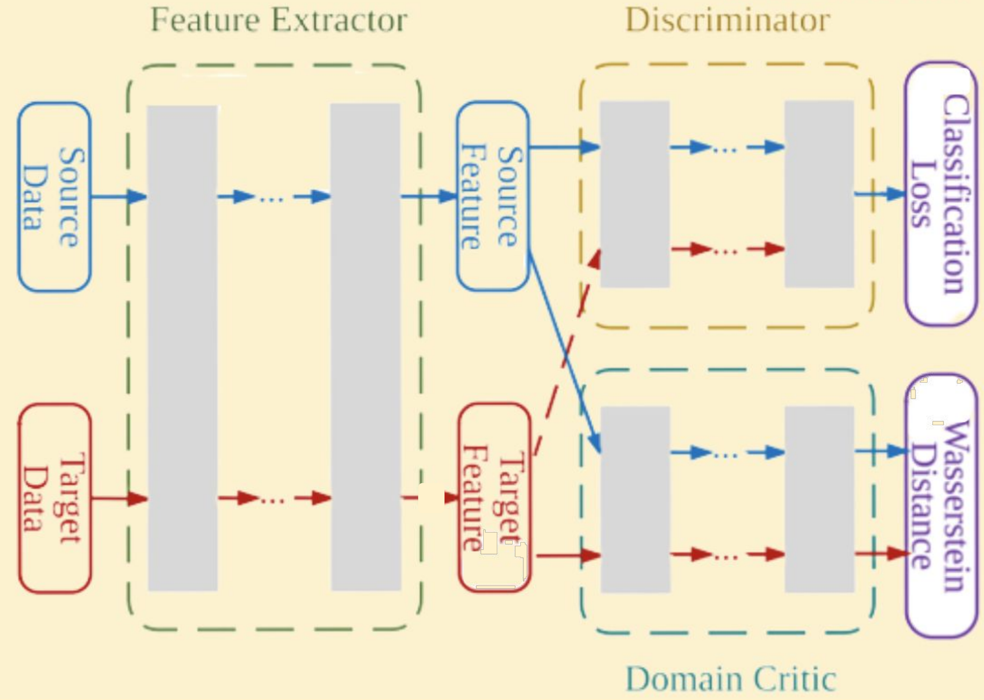
- Significant gap between empty squares and pawns
- Data generation and the preprocessing pipeline took time to R&D
- Not invariant to piece set / board texture
- High computational cost

Want to pick up on our work?

Alternative 1: Deep-Reconstruction Classification Network (DRCN)[10]



Alternative 2: Using Wasserstein Distance Guided Representation Learning Model (WDGRL)[11]



[10]: M. Ghifary et al., "Deep reconstruction-classification networks for unsupervised domain adaptation," in Proc. Eur. Conf. Comput. Vis. (ECCV), pp. 597-613, 2016.
[11]: J. Shen et al., "Wasserstein distance guided representation learning for domain adaptation," in Proc. AAAI Conf. Artif. Intell., vol. 32, no. 1, 2018.

References

[1] S. Ben-David et al., "A theory of learning from different domains," Mach. Learn., vol. 79, pp. 151-175, 2010.

[2] A. D. S. D. Neto and R. M. Campello, "Chess position identification using pieces classification based on synthetic images generation and deep neural network fine-tuning," in Proc. 21st Symp. Virtual Augmented Reality (SVR), pp. 152-160, IEEE, Oct. 2019.

[3] Elucidation, "Elucidation/chessboarddetect: Hodgepodge of chessboard detection algorithms on images from actual matches.," GitHub. [Online]. Available: https://github.com/Elucidation/ChessboardDetect. [Accessed: 10-Apr-2023].

[4] "The best free, adless chess server," lichess.org. [Online]. Available: https://lichess.org/. [Accessed: 10-Apr-2023].

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, 'Focal Loss for Dense Object Detection', arXiv [cs.CV]. 2018.

[6] B. Sun, J. Feng, and K. Saenko, 'Correlation Alignment for Unsupervised Domain Adaptation', CoRR, vol. abs/1612.01939, 2016.

[7] K. Simonyan and A. Zisserman, 'Very Deep Convolutional Networks for Large-Scale Image Recognition', CoRR, vol. abs/1409.1556, 2014.

[8] B. Sun, J. Feng, and K. Saenko, 'Correlation Alignment for Unsupervised Domain Adaptation', CoRR, vol. abs/1612.01939, 2016.

[9] Y. Ganin et al., 'Domain-Adversarial Training of Neural Networks', arXiv [stat.ML]. 2016.

[10]: M. Ghifary et al., "Deep reconstruction-classification networks for unsupervised domain adaptation," in Proc. Eur. Conf. Comput. Vis. (ECCV), pp. 597-613, 2016.

[11]: J. Shen et al., "Wasserstein distance guided representation learning for domain adaptation," in Proc. AAAI Conf. Artif. Intell., vol. 32, no. 1, 2018.

Questions?