SATELLITE IMAGERY RECOGNITION

PROJECT I

Under the guidance of Dr. Shibaprasad Sen

By
VIVEK THAKUR
HARSH RANJAN
DIVYA KUMAR BAID
SUMEDHA SARKAR
RITAM KONAR

B.Tech
CSE (Artificial Intelligence and Machine Learning)

CONTENT

- Problem Analysis
- Benchmark and Assumption
- Approaches
- Data design
- Architectural design
- Procedural design
- Result
- Evaluation
- Conclusion

PROBELM ANALYSIS

Satellite Imagery Recognition Challenges:

- Growing Importance: Vital for military surveillance, disaster monitoring, and urban planning.
- Complexity in Detection: Overhead imagery poses challenges, especially in detecting small objects.
- Deep Learning Struggle: Traditional techniques face hurdles in adapting deep learning for aerial object detection.
- Key Hurdles: Small object size, scale variations, illumination issues, orientation diversity, and limited annotated datasets.
- Objective: Overcoming these challenges for effective and accurate satellite imagery analytics.

PROBELM ANALYSIS

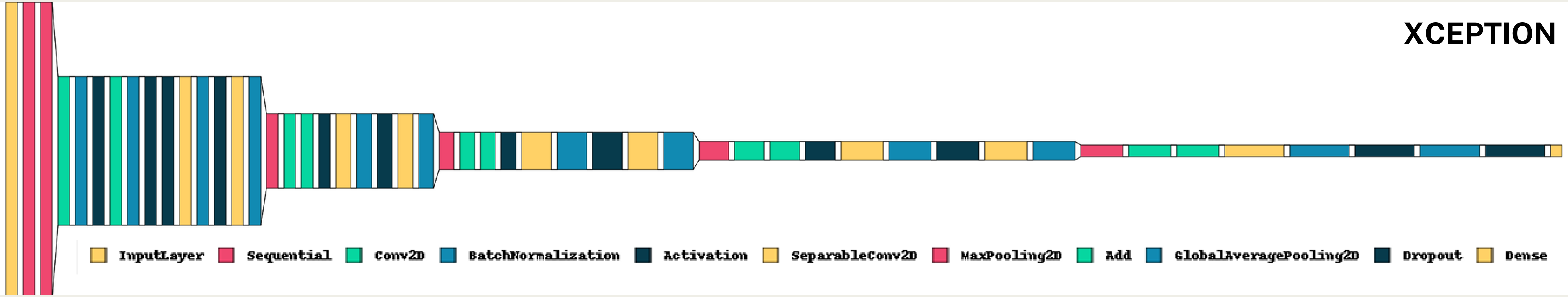
- Dataset Source: DARPA and U.S. Air Force's MSTAR Dataset (1995-1997) on Kaggle.
- Nature of Dataset: 9 classes via synthetic aperture radar (SAR) for cloud-penetrating vehicle identification.
- Models:
  - MLP-ANN Model: Initial exploration.
  - Simple & Deeper CNNs: Evolved for feature-rich SAR image analysis.
- Refinement:
  - Data Augmentation: Expand dataset artificially.
  - Dropout Layers: Prevent overfitting.
- Callbacks:
  - Early Stopping: Halts training at performance plateaus.
  - Learning Rate Reduction: Optimizes convergence.
- Objective: Optimal vehicle identification in SAR images through iterative model evolution and strategic refinements.

BENCHMARK AND ASSUMPTION

- Data collected using one-foot resolution spotlight mode X-band SAR sensor.
- Clutter data collected in strip map mode at 1-foot resolution.
- Spotlight mode utilized for data collection at depression angles of 15, 17, 30, and 45 degrees.

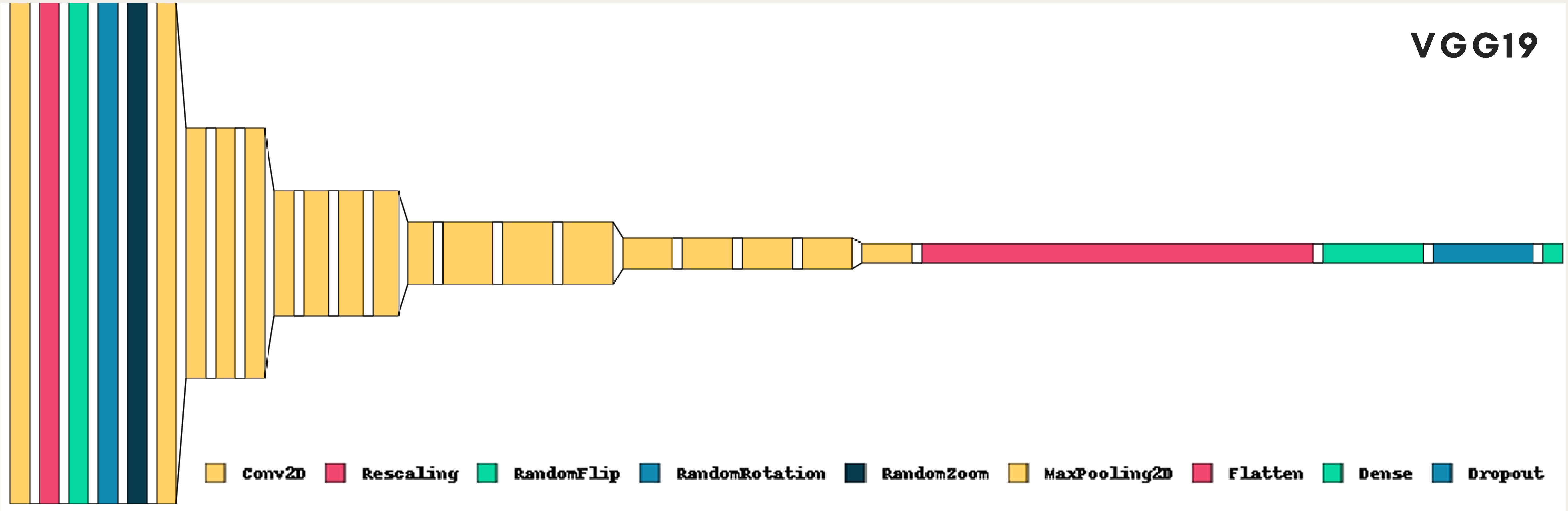
- **Dataset split**: 60% training, 20% validation, 20% test.
- **Hyperparameter tuning** performed on the validation set.
- Final model selected based on <u>F2 score, accuracy, and Matthews Correlation.</u>
- Testing conducted on reserved data.
- **Objective**: Achieve high accuracy and recall.
- **Accuracy**: Model's ability to identify different targets accurately.
- **Recall**: Model's capability to capture all pertinent results effectively.

BENCHMARK MODELS



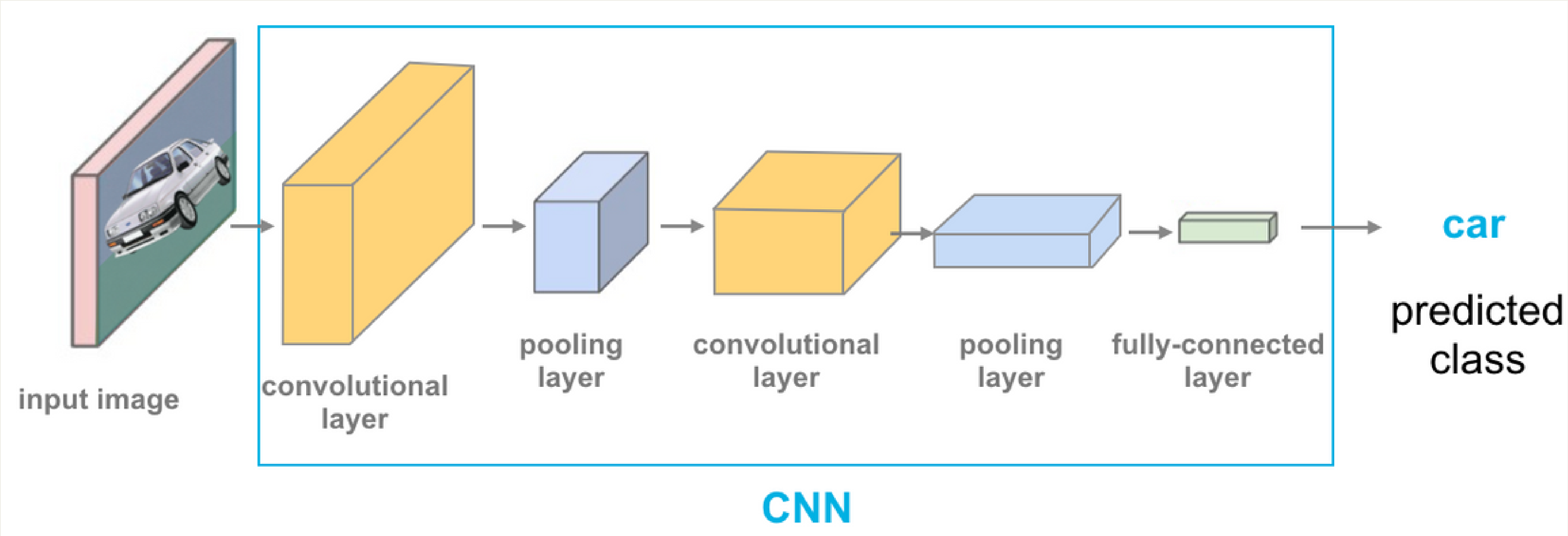
Xception, known for its depth and feature extraction prowess, employs depth-wise separable convolutions, making it a potent choice for transfer learning and benchmark dataset tasks.

BENCHMARK MODELS



VGG19, with its 19-layer architecture featuring convolutional and max-pooling layers, excels at feature extraction in image-related tasks like classification and object recognition.

APPROACH



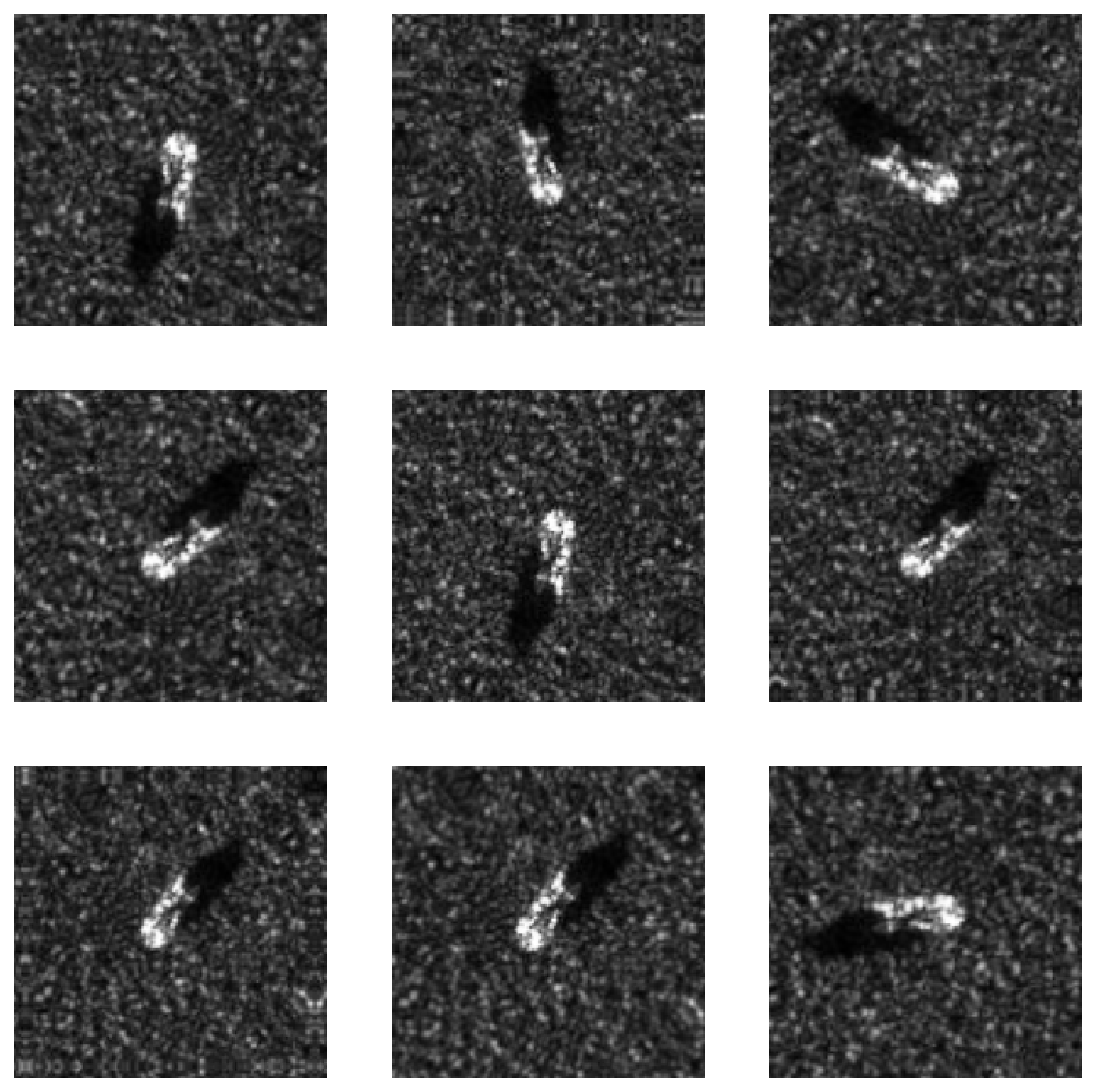
CNN for Image Classification:

- **Objective**: Efficient image classification.
- **Capability**: Identifies spatial patterns in 3D image space.
- **Contrast** Differs from traditional networks, focusing on pixel groups.
- **Functionality**: Nodes process input, generating meaningful outputs.
- **Diversity**: Utilizes various layers, emphasizing convolutional, fully-connected and pooling for enhanced feature extraction.

APPROACH

Data Augmentation:

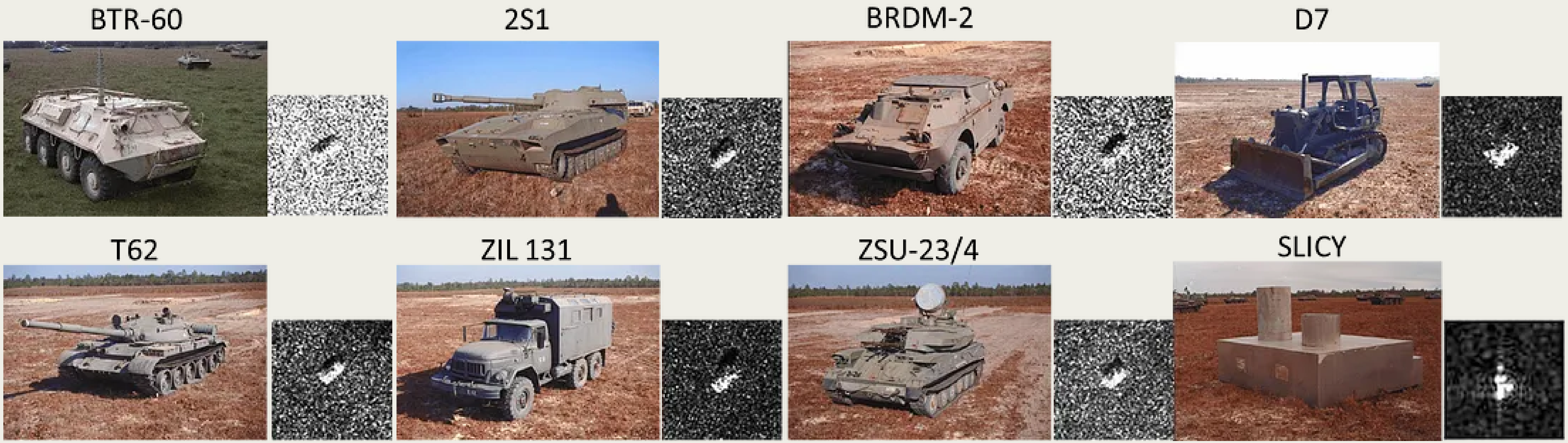
- **Objective**:Enhance dataset diversity and generalization.
- **Techniques Used**:
  - Flipping
  - Rotating
  - Zooming
- **Purpose**: Mitigate overfitting by generating varied data orientations.
- **Benefits**:
  - Addresses limited data availability.
  - Improves model robustness.
- **Implementation**: Applied flipping, rotating, and zooming to augment dataset.



APPROACH

- **Early stopping** is an optimization technique used to reduce overfitting without compromising on model accuracy. The main idea behind early stopping is to stop training before a model starts to overfit.
- **Learning Rate Reduction** at Plateau: Optimizing the learning rate based on performance plateaus, enhancing the model's ability to reach new minima and improve results.
- For **image classification**, we explored models detecting multiple objects, aiming to identify the best classifier through iterations and finalize the evaluation with a **holdout** dataset.

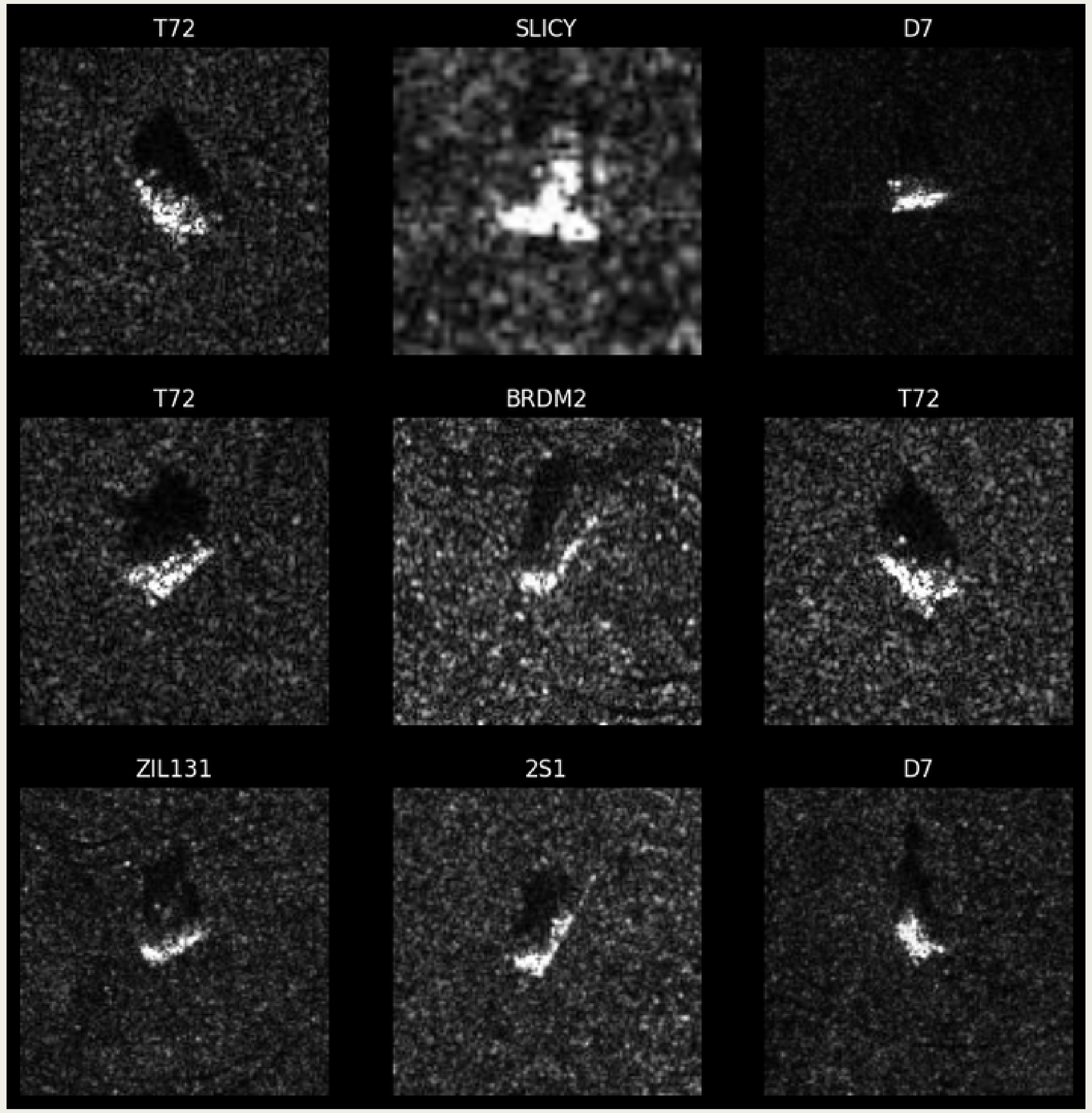
DATA DESIGN



Vehicle classes (left), satellite images of same classes (right)

Vehicle images were captured using synthetic aperture radar (SAR), a satellite imagery form that penetrates atmospheric conditions like clouds. The images don't visually reveal the contained vehicle.

DATA DESIGN



The first 9 images of our training data and the corresponding labels

The data contains the following 9 Classes:

- **2S1 Gvozdika** – Self-propelled artillery
- **ZSU-23-4 Shilka** – Self-propelled anti-aircraft
- **BRDM-2** – Amphibious armored scout car
- **BTR-60** – Armored personnel carrier
- **D7** – Caterpillar Bulldozer
- **ZIL-131** – Military cargo truck
- **T-62** – Main battle tank
- **T-72** – main battle tank (2nd Gen)
- **SLICY** – Structure acting as a 'ground truth' (not a vehicle)

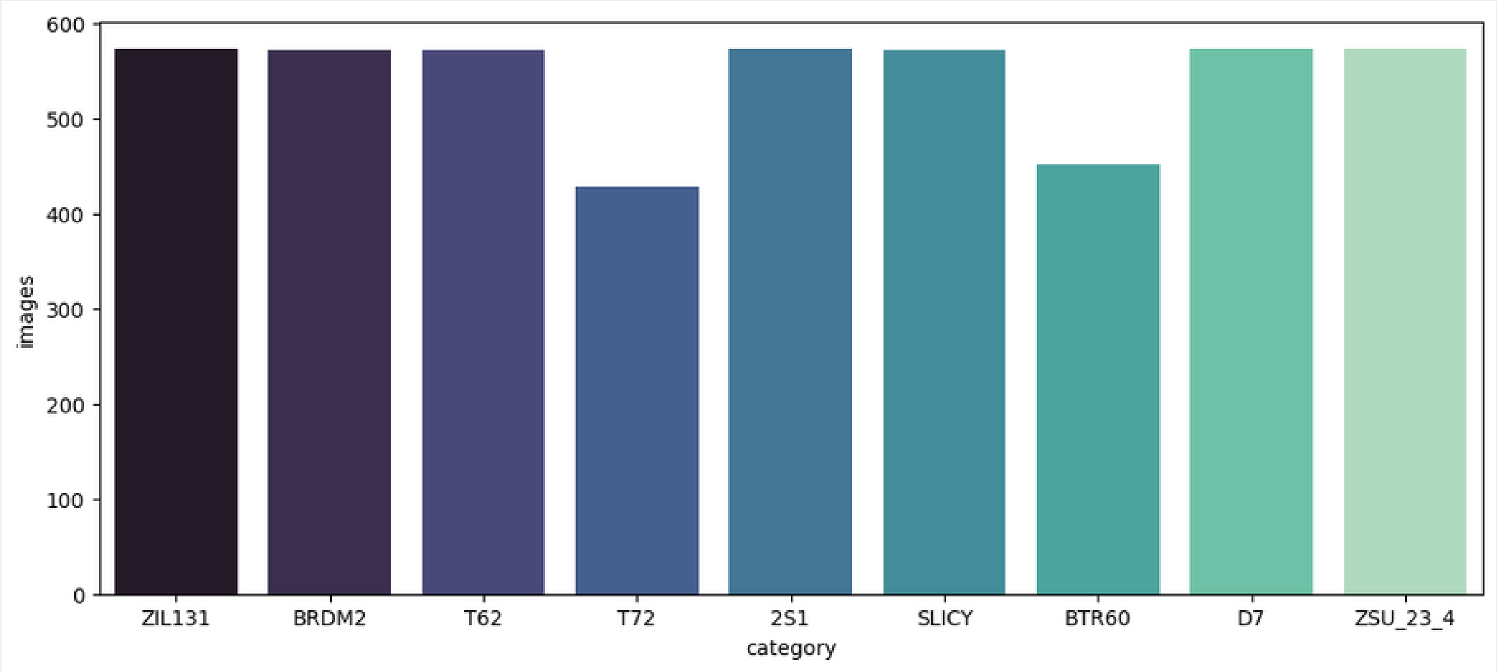
- Found **4887** files belonging to **9** classes.
- Using **3910** files for training.
- Using **977** files for validation.

- Number of validation batches: **25**
- Number of test batches: **6**

DATA DESIGN

DATA PREPROCESSING AND EVALUATION FUNCTIONS

- **def plot_total_images(base_dir)** : to plot Simple barplot on the total number of images present in the dataset
- **def plot_metrics(history)** : Function for plotting selected metrics across model epochs. We have used loss, precision-recall curve, F2, and Matthews Correlation Coefficient, but any metric TensorFlow or TensorFlow addons can be used.

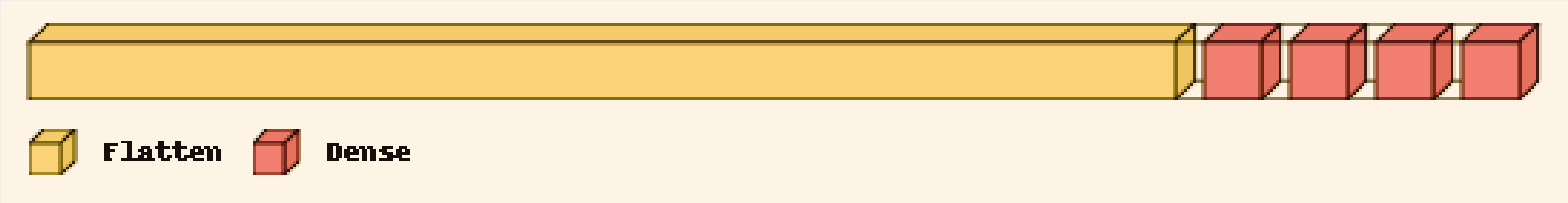


- **def plot_total_images(base_dir) :** to plot Simple barplot on the total number of images present in the dataset
- **def plot_metrics(history) :** Function for plotting selected metrics across model epochs. We have used loss, precision-recall curve, F2, and Matthews Correlation Coefficient, but any metric TensorFlow or TensorFlow addons can be used.

ARCHITECTURAL DESIGN

ANN-MLP

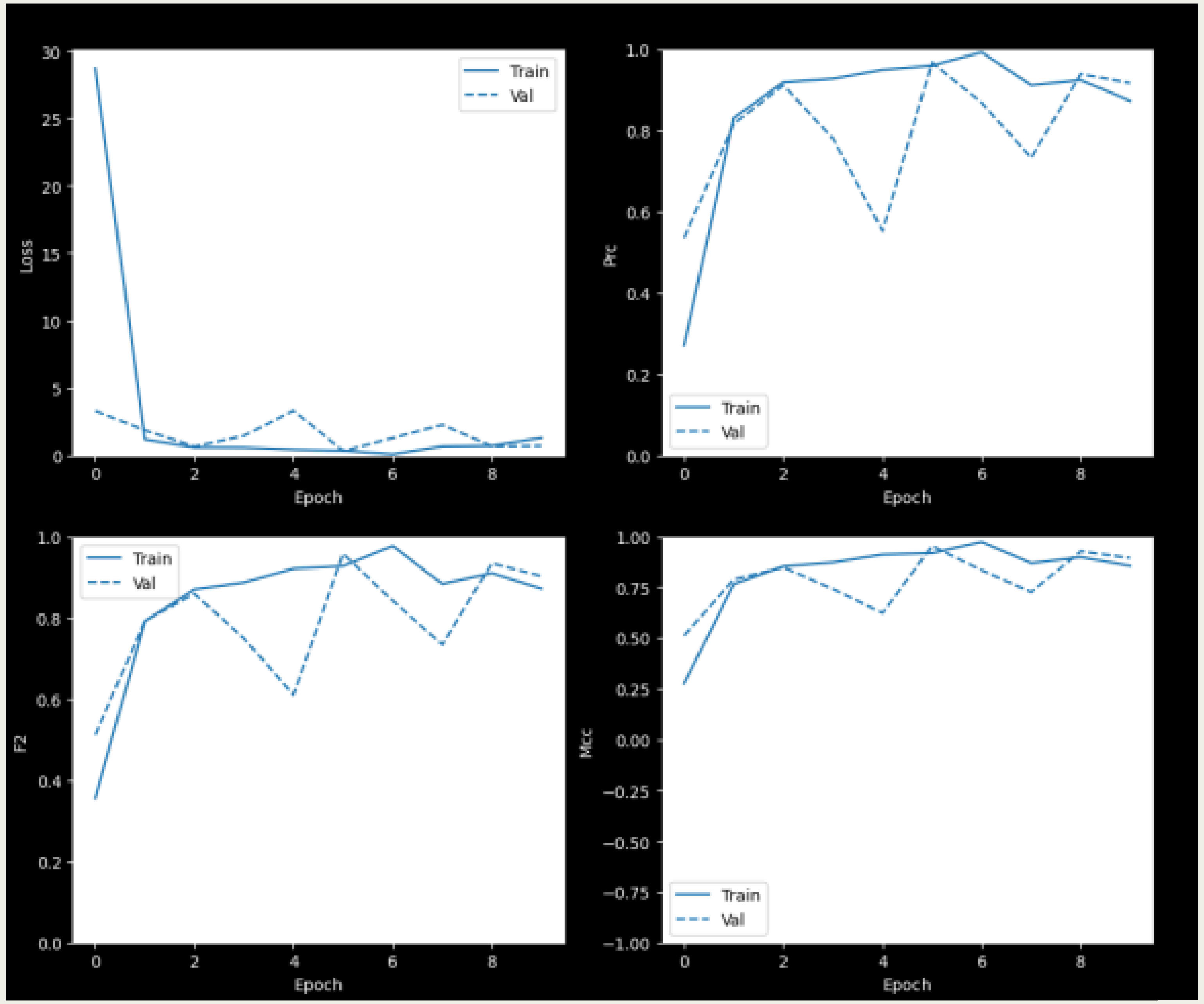
As a starting point, We built a simple multi-layer perceptron (MLP) model, which is just a fancy way of saying it's a model with fully-connected layers. We set each of our three layers to have 100 nodes, and used a Rectified Linear Unit (ReLU) activation function.



Layers of convolution model for ANN-MLP

RESULTS & EVALUATION

ANN-MLP



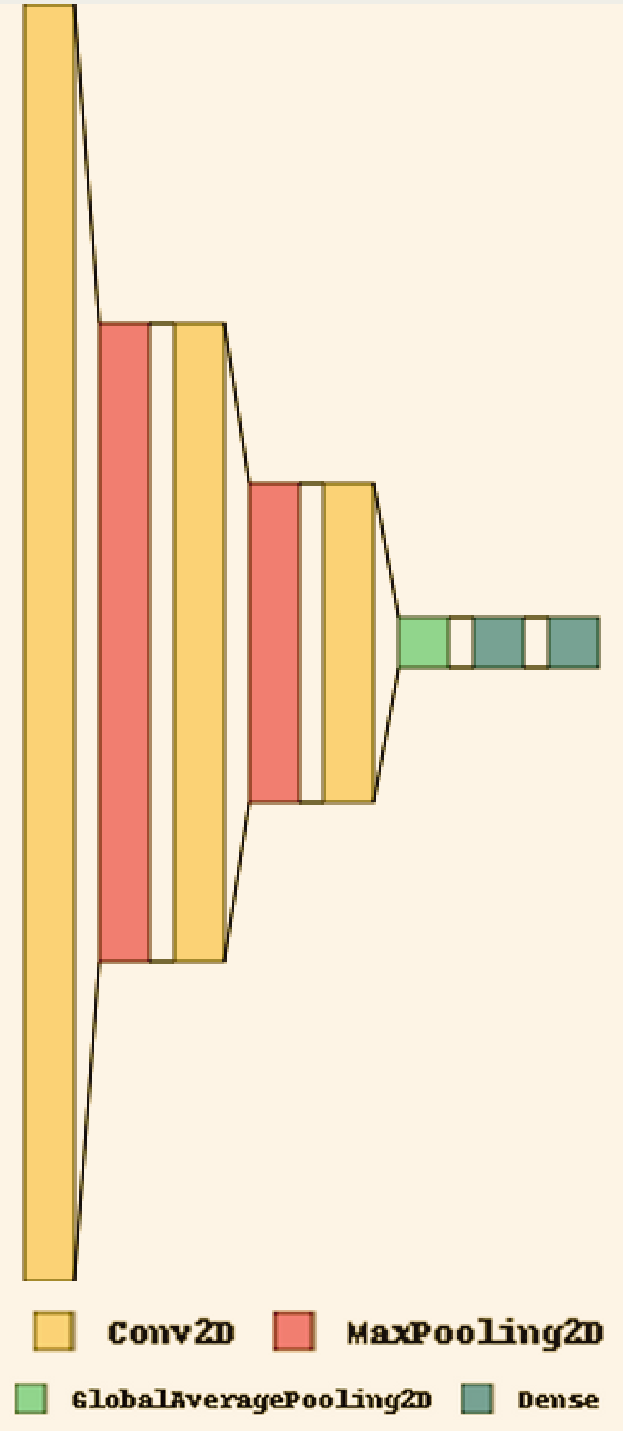
Holdout Results

{
'loss': 0.9413747191429138,
'categorical_accuracy': 0.890625,
'MCC': 0.8783231377601624,
'F2': 0.8886363506317139,
'auc': 0.9662576913833618,
'prc': 0.8978415131568909
}

THIS IMAGE MOST LIKELY BELONGS TO T62 WITH A 25.36 PERCENT CONFIDENCE.

ARCHITECTURAL DESIGN

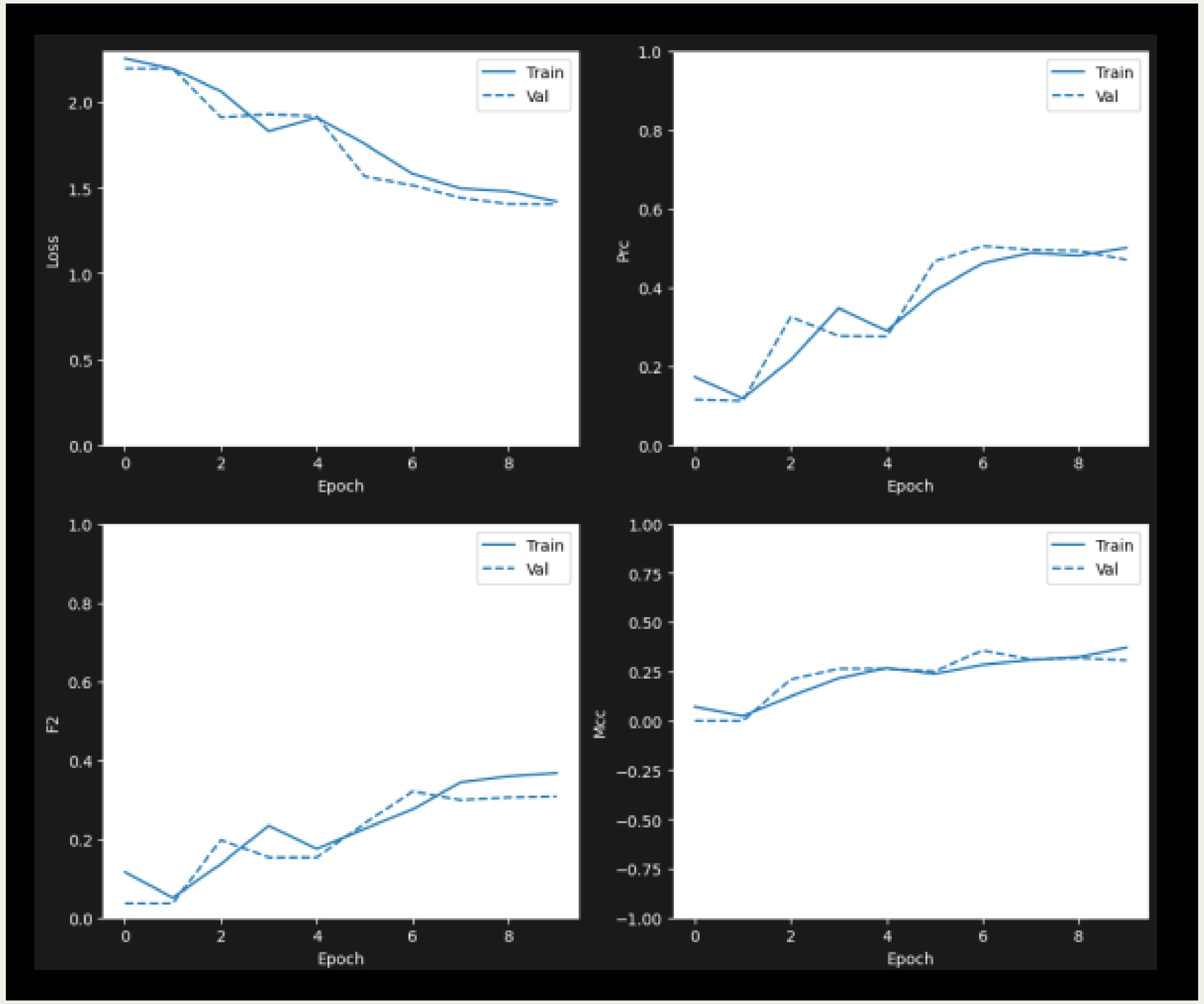
MODEL 1



Our model is a Sequential architecture that includes Conv2D and MaxPooling2D layers for feature extraction. The model consists of Conv2D and MaxPooling2D layers for feature extraction, followed by GlobalAveragePooling2D for dimensionality reduction. It concludes with two Dense layers for prediction.

RESULTS & EVALUATION

MODEL 1



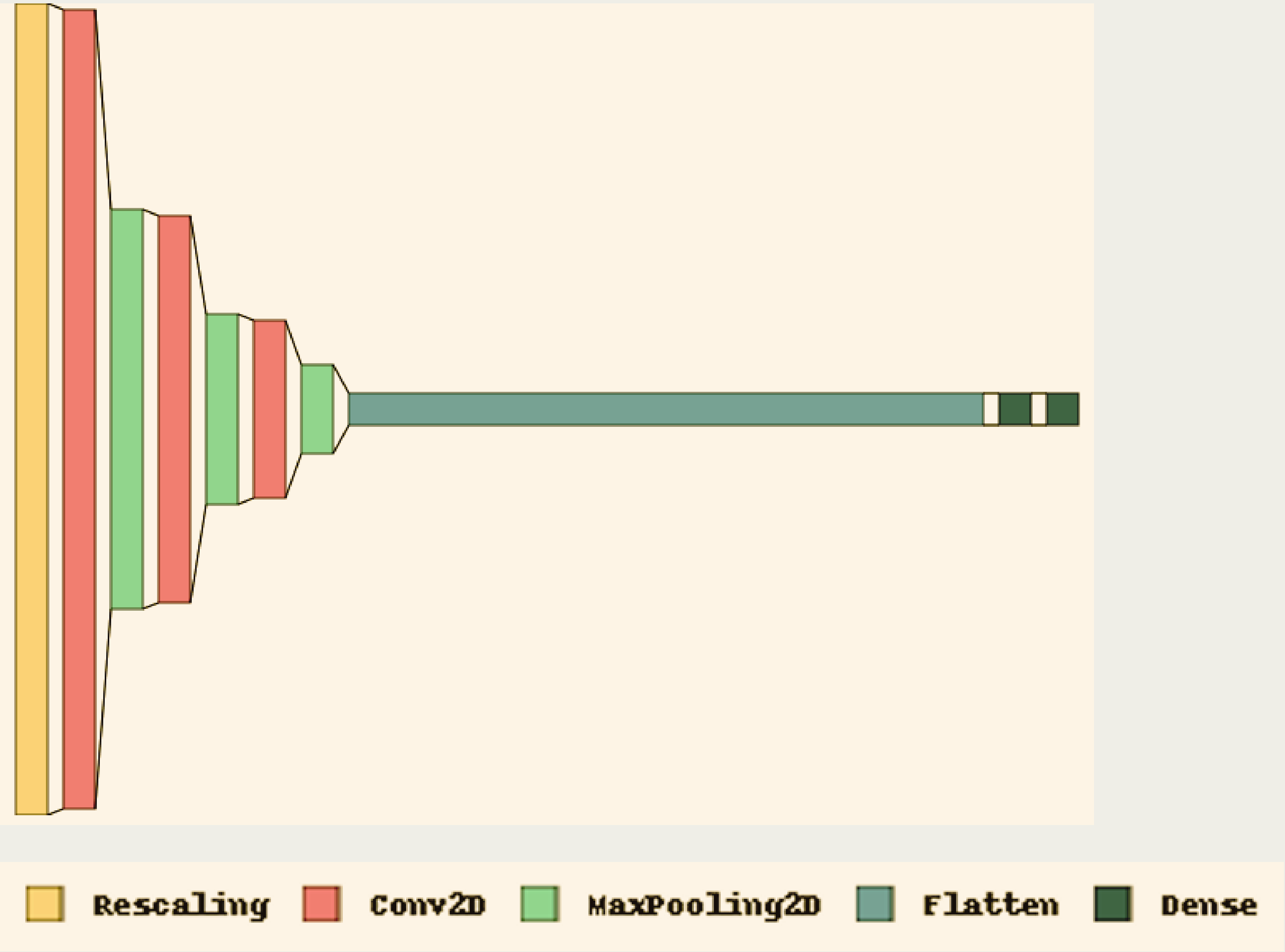
Holdout Results

{
'**loss**': 1.353049397468567,
'**categorical_accuracy**': 0.3854166,
'**MCC**': 0.3362288475036621,
'**F2**': 0.3391103744506836,
'**auc**': 0.8748660683631897,
'**prc**': 0.4999695420265198
}

THIS IMAGE MOST LIKELY BELONGS TO D7 WITH A 14.23 PERCENT CONFIDENCE.’

ARCHITECTURAL DESIGN

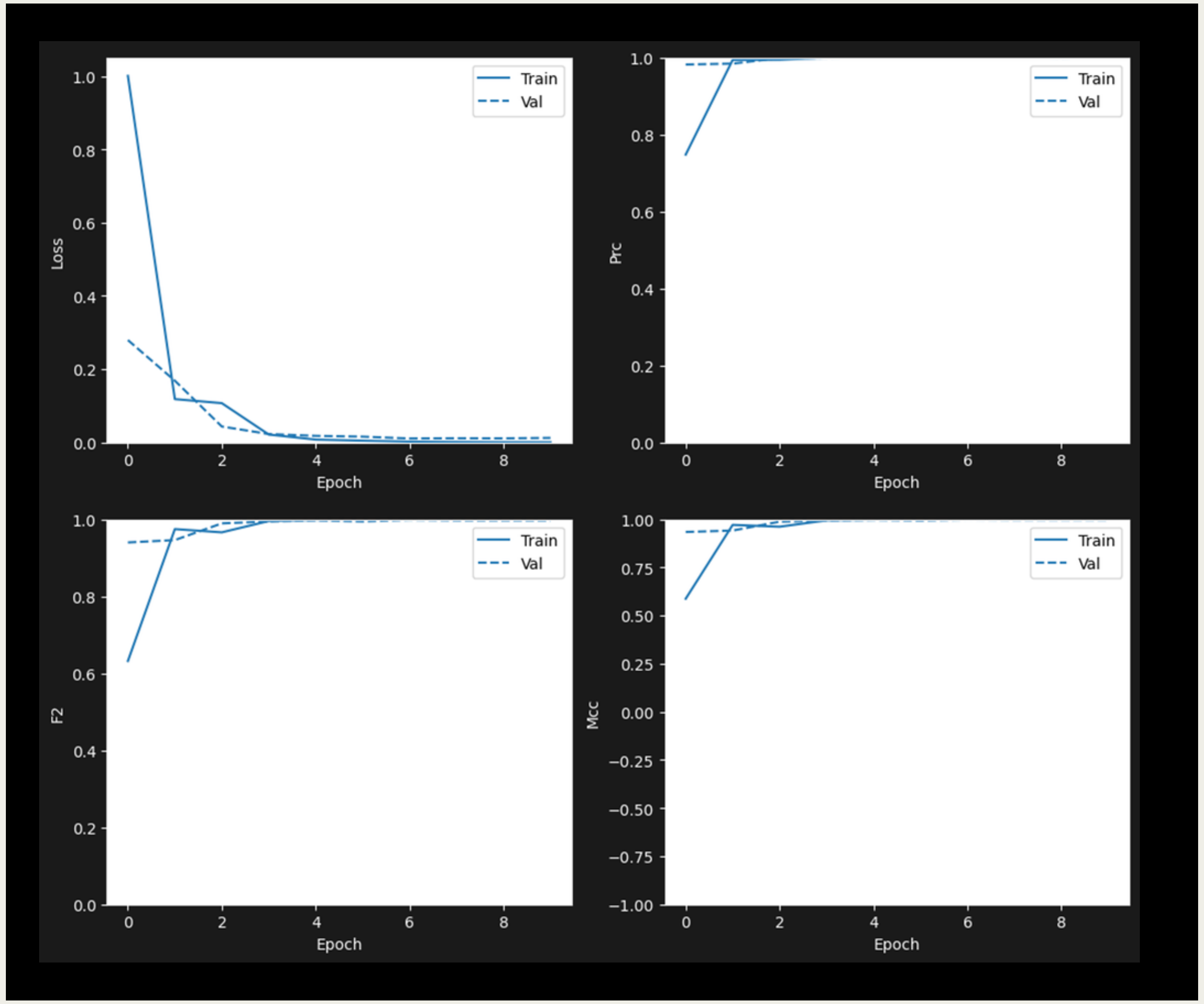
MODEL 2



Experimented with a convolutional neural network (CNN) for image object detection. CNNs are ideal for this task as they utilize filters to scan an image iteratively, extracting important features. These filters progress from identifying simple shapes to more complex objects.

RESULTS & EVALUATION

MODEL 2



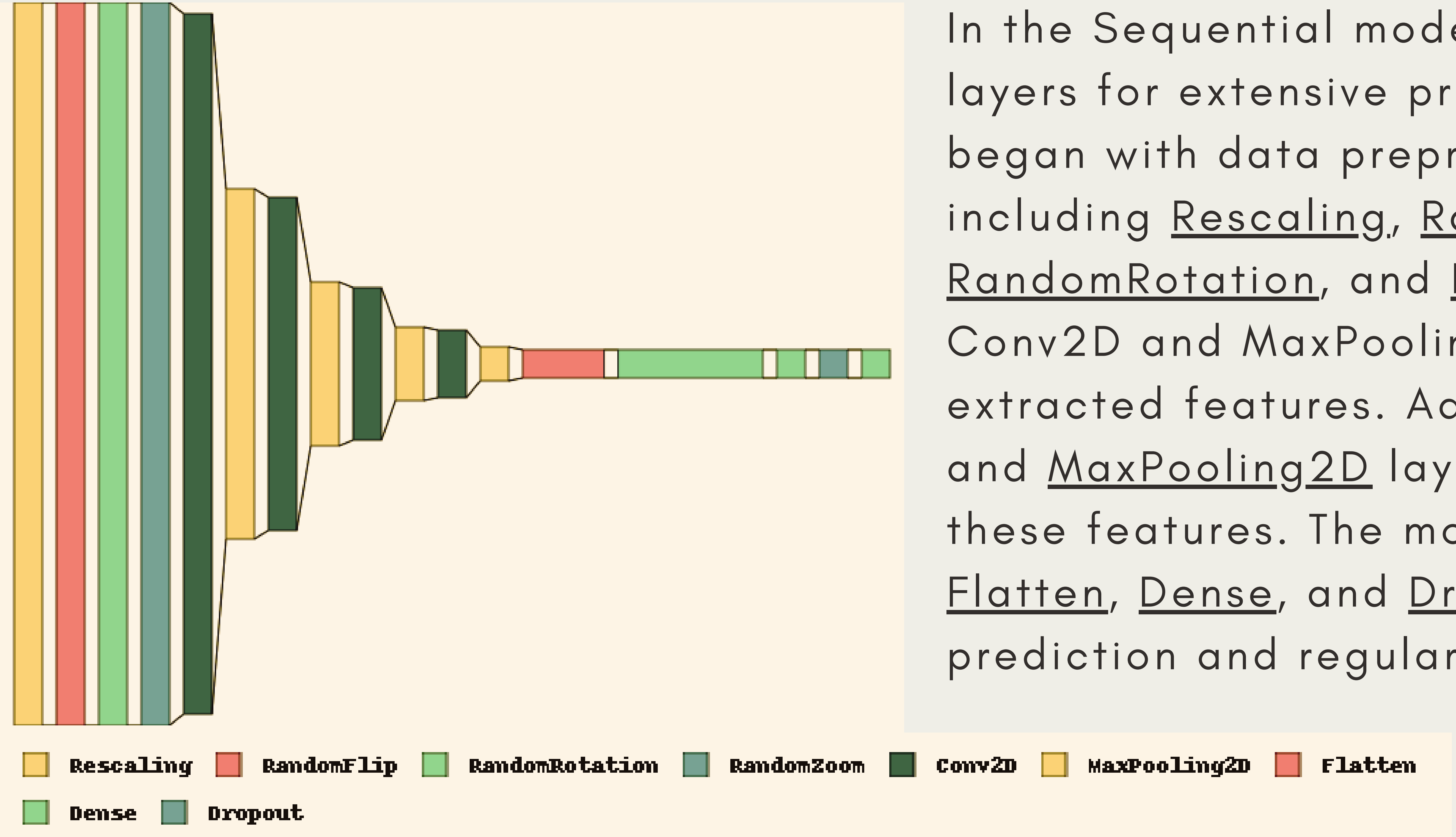
Holdout Results

{
'**loss**': 0.017901975661516l9,
'**categorical_accuracy**': 0.994791686,
'**MCC**': 0.9941287636756897,
'**F2**': 0.9947527647018433,
'**auc**': 0.9999797344207764,
'**prc**': 0.9998376369476318
}

THIS IMAGE MOST LIKELY BELONGS TO T72 WITH A 25.36 PERCENT CONFIDENCE.

ARCHITECTURAL DESIGN

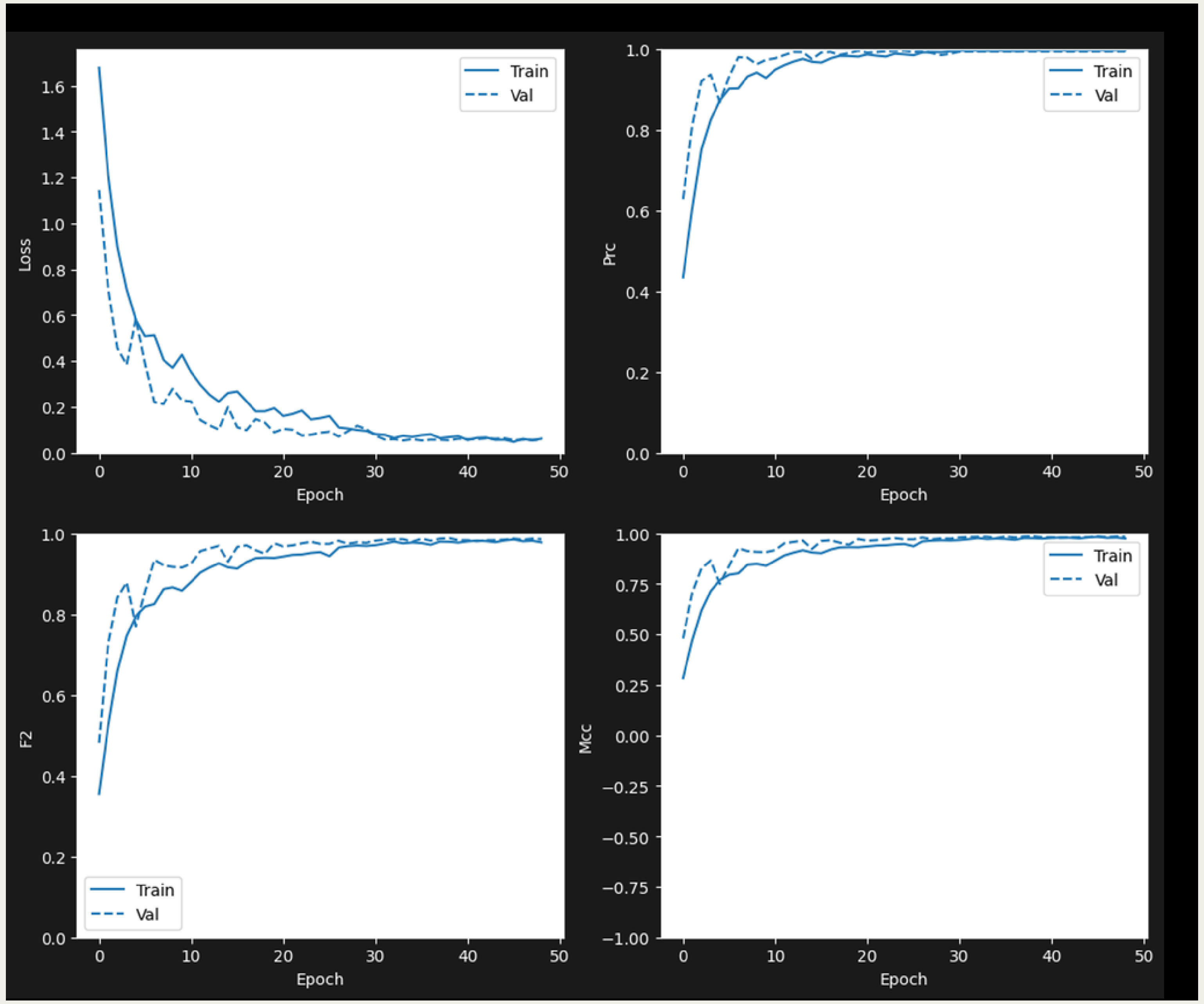
MODEL 3: REFINED CNN



In the Sequential model, we used various layers for extensive processing. We began with data preprocessing layers, including Rescaling, RandomFlip, RandomRotation, and RandomZoom. Next, Conv2D and MaxPooling2D layers extracted features. Additional Conv2D and MaxPooling2D layers further refined these features. The model concluded with Flatten, Dense, and Dropout layers for prediction and regularization.

RESULTS & EVALUATION

MODEL 3

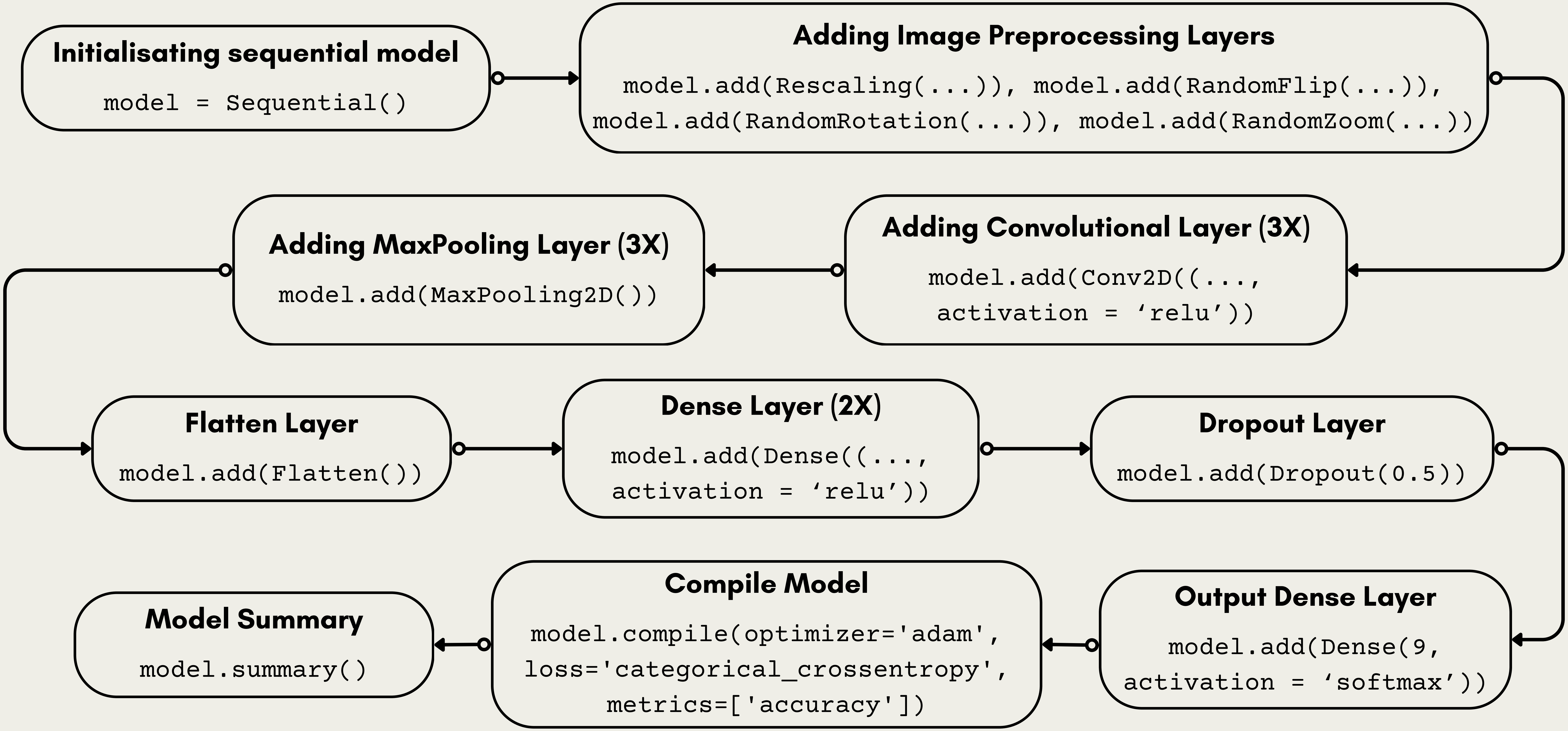


Holdout Results

{
'**loss**': 0.014037217013537884,
'**categorical_accuracy**': 0.99479168,
'**MCC**': 0.9941301345825195,
'**F2**': 0.9947640299797058,
'**auc**': 0.9999966025352478,
'**prc**': 0.9999729990959167
}

THIS IMAGE MOST LIKELY BELONGS TO T72 WITH A 25.36 PERCENT CONFIDENCE.

PROCEDURAL DESIGN



MAXIMUM CONFIDENCE PERCENTAGE

- **percentage_of_files_for_T72** = 428 / 4887 * 100% ≈ 8.76%
- **normalized_batch_size_for_T72** = 30 / 977 * 100% ≈ 3.08%
- **weighted_contribution_of_T72** = 8.76% * 3.08 ≈ 0.27%
- **total_softmax_score** = 0.27% + 0.28% + 0.23% + 0.27% + 0.28% + 0.28% + 0.21% + 0.27% + 0.27% ≈ 2.56%
- **average_softmax_score_per_class** = 2.56% / 9 * 100% ≈ 28.44%
- **expected_softmax_score_for_T72** = 28.44 * 8.76% ≈ 2.50%
- **confidence_percentage_for_T72** = 2.50% / 2.56% * 100% ≈ 100%

MAXIMUM CONFIDENCE PERCENTAGE

Maximum Confidence Percentage (25.36%):

- This is calculated as the difference between the **average_softmax_score_per_class** and **normalized_batch_size_for_T72**: 28.44% - 3.08% = 25.36%

- Relation to Maximum Confidence Percentage:
    - The calculated maximum confidence percentage (25.36%) aligns with the expected softmax score for class T72, indicating that the model has high confidence in predicting T72 in this particular instance.

- In summary, the analysis provides insights into the contribution of class T72 to the overall softmax score and the resulting confidence percentage, highlighting its significance in the model's predictions.

COMPARATIVE STUDY

This table provides an overview of the performance metrics for each model, allowing for a clear comparison.

Model	Categorical Accuracy	Matthews Correlation Coefficient	F2 Score	ROC-AUC	Precision-Recall Curve	Loss
Xception	0.78125	0.7666	0.7605	0.9683	0.8820	0.7668
VGG19	0.90104	0.8893	0.8988	0.9941	0.9622	0.3926
Our CNN Model	0.99479	0.9941	0.9948	0.9999	0.9997	0.0140

These findings underscore the importance of model selection in achieving superior results, with Our CNN Model standing out as the most effective solution for the given objective.

CONCLUSION

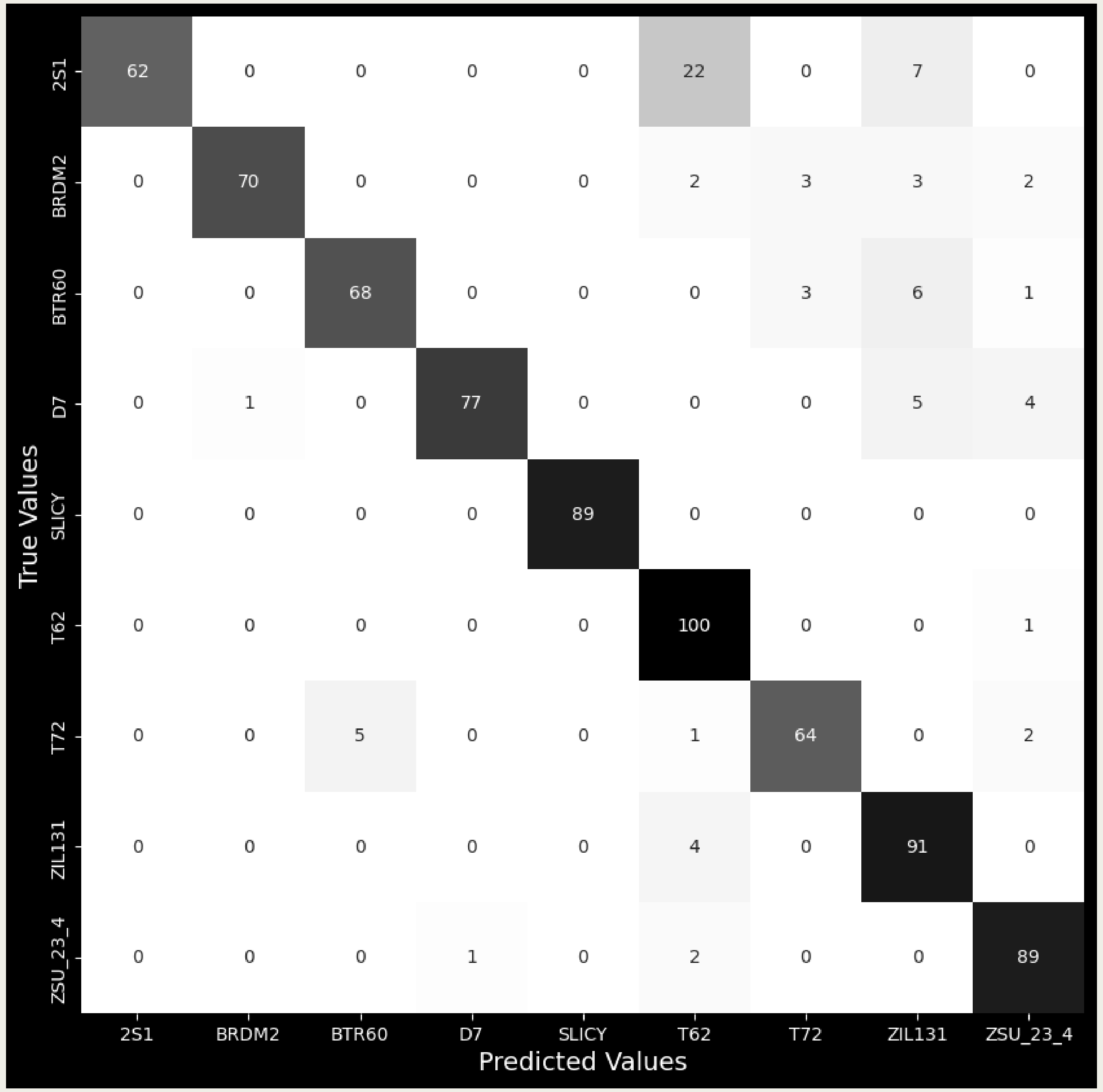
- Our models excel in real-world military vehicle detection, particularly with SAR data, underscoring the significance of machine learning and computer vision for this field's progress.

- **Ethics & social responsibilities**: Ethical principles guided our project, addressing misuse, biases, and privacy. We emphasized responsible data science through collaboration and restricted our model's applicability to contemporary imagery, committing to ethical use.

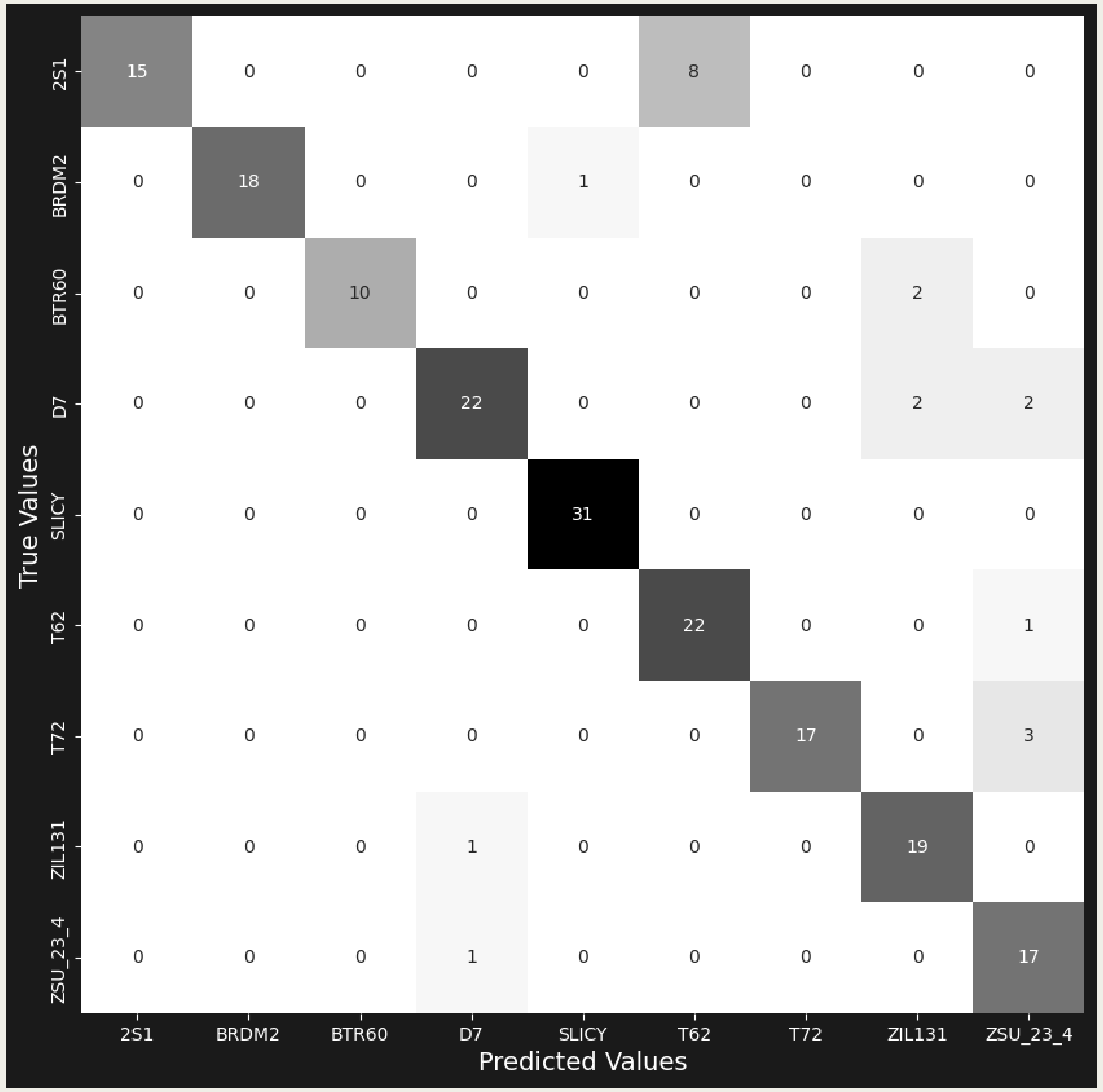
- **Future opportunities** in this field involve diversifying the dataset, using synthetic data, trying alternative architectures, cross-platform testing, accounting for data variability, factoring in environmental conditions, and addressing challenges to enhance military applications and adapt to associated issues.

APPENDIX

ANN - MLP

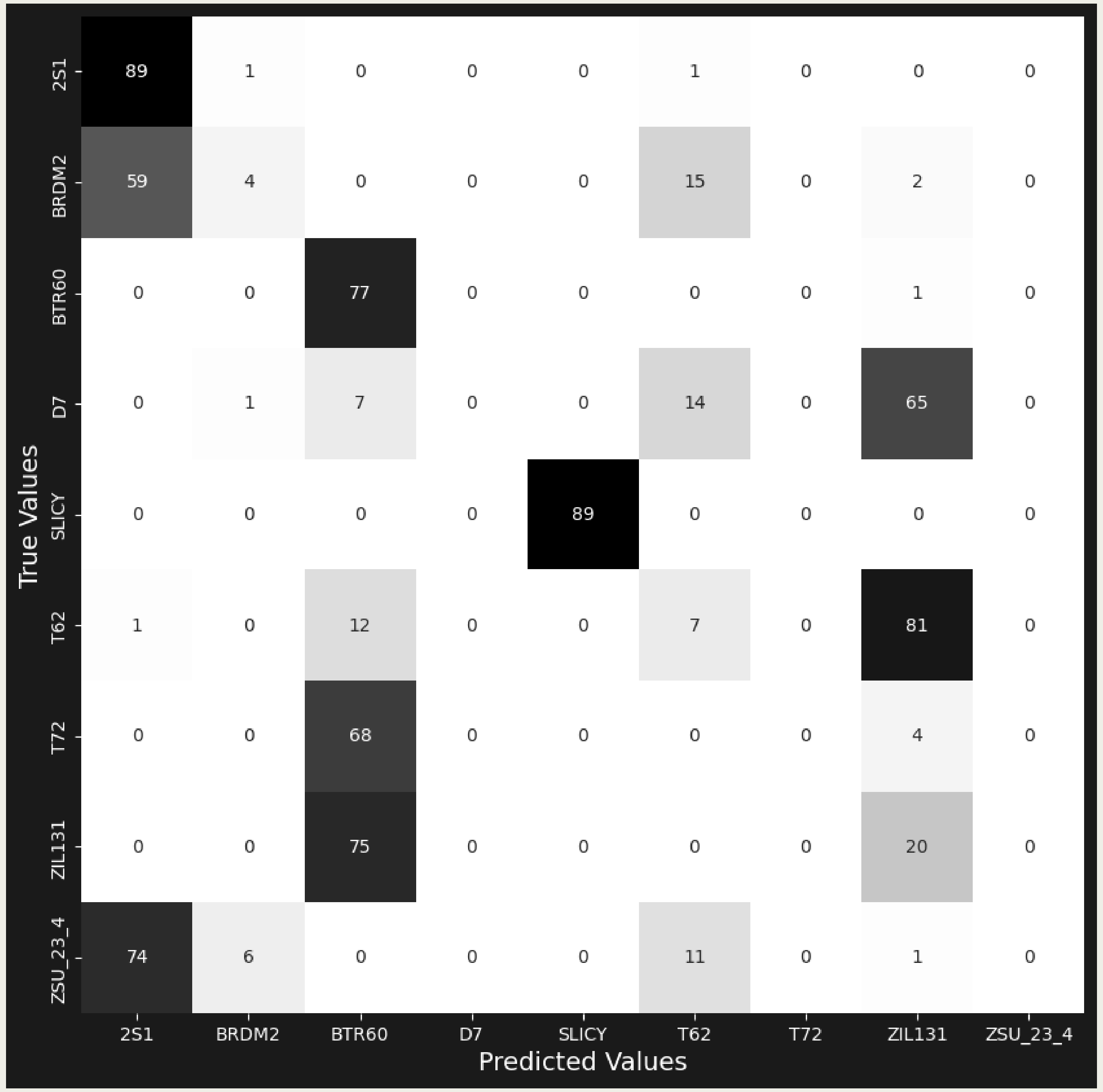


Confusion Matrix for Validation dataset

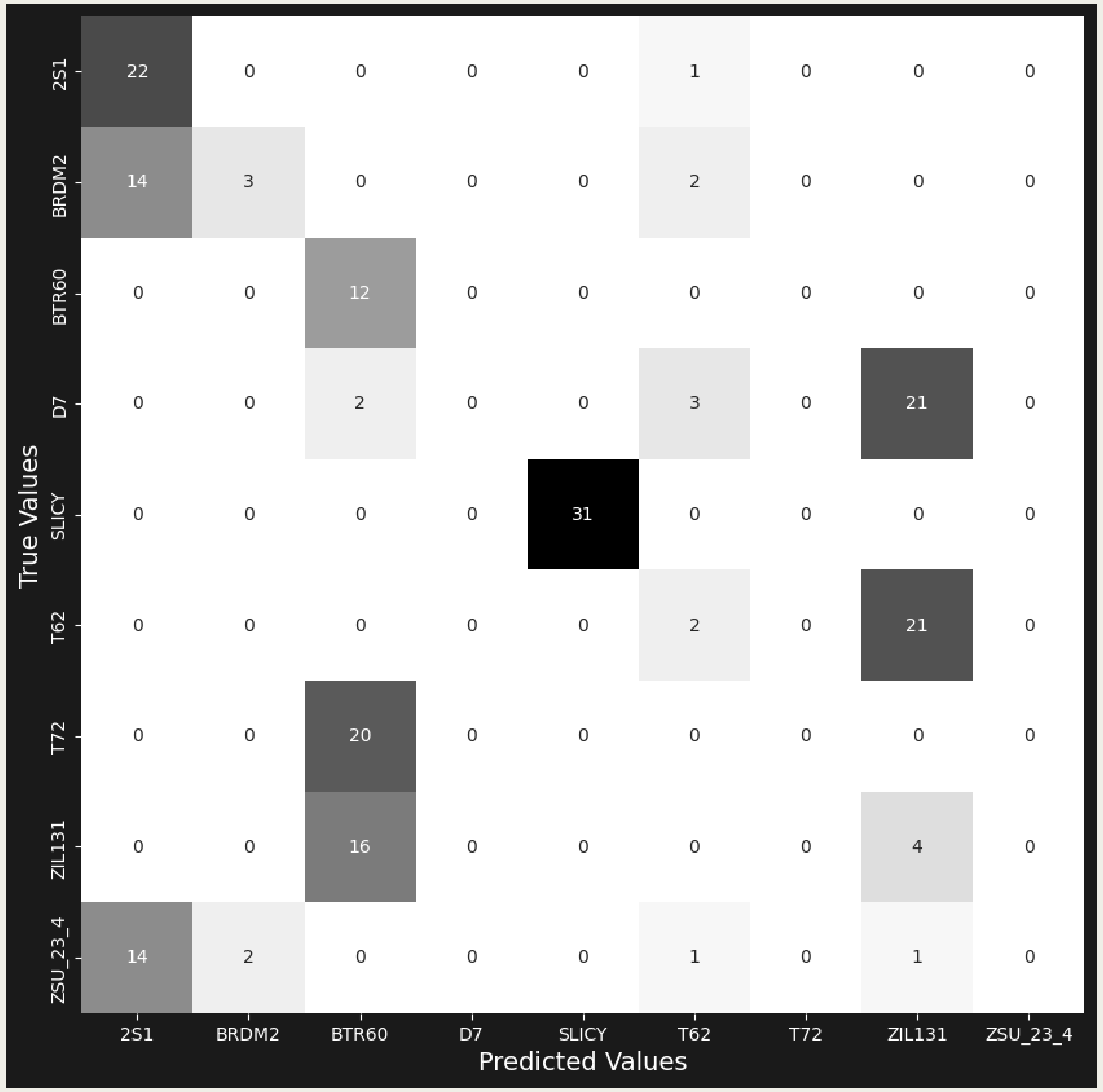


Confusion Matrix for Testing dataset

MODEL 1

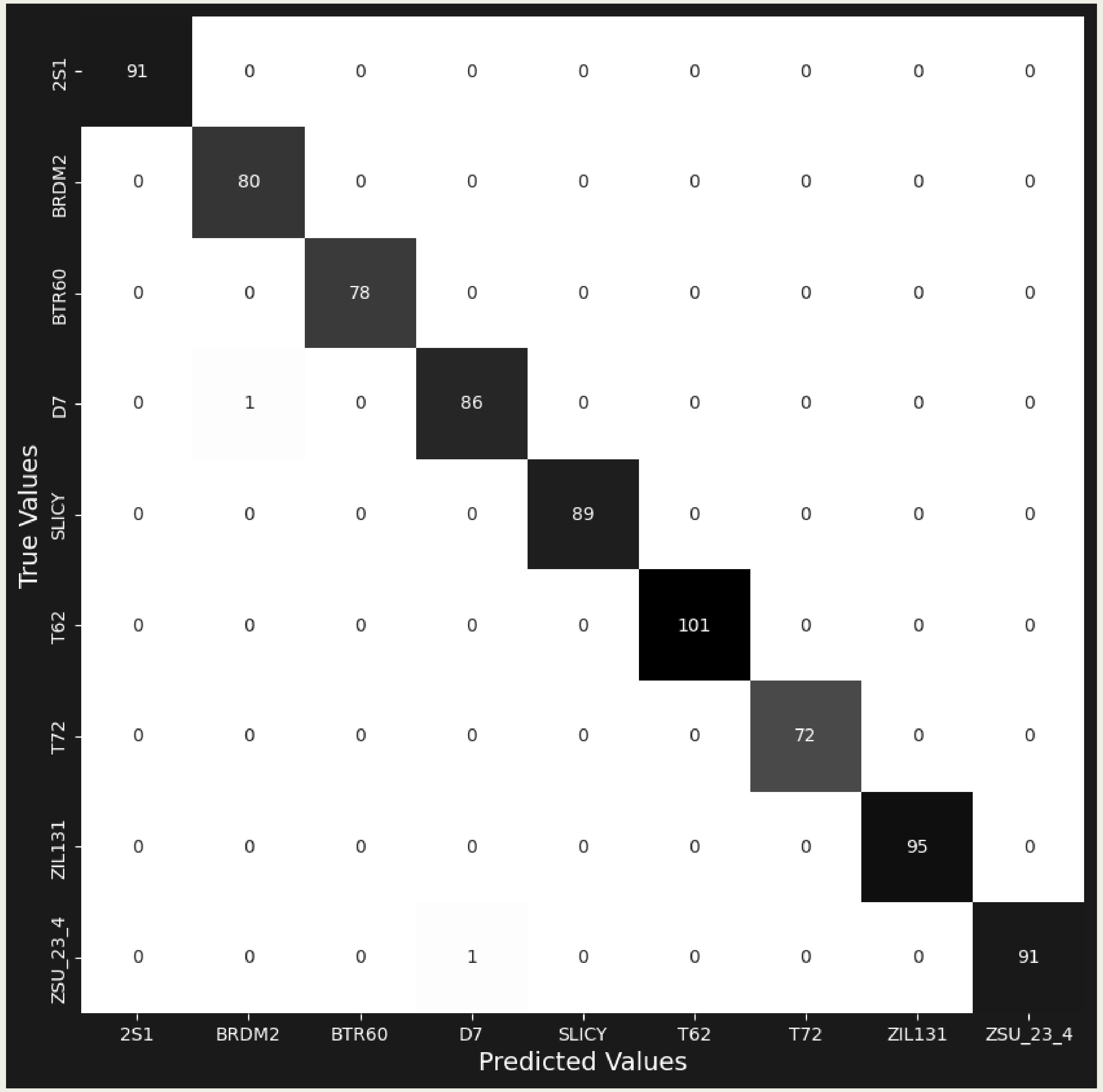


Confusion Matrix for Validation dataset

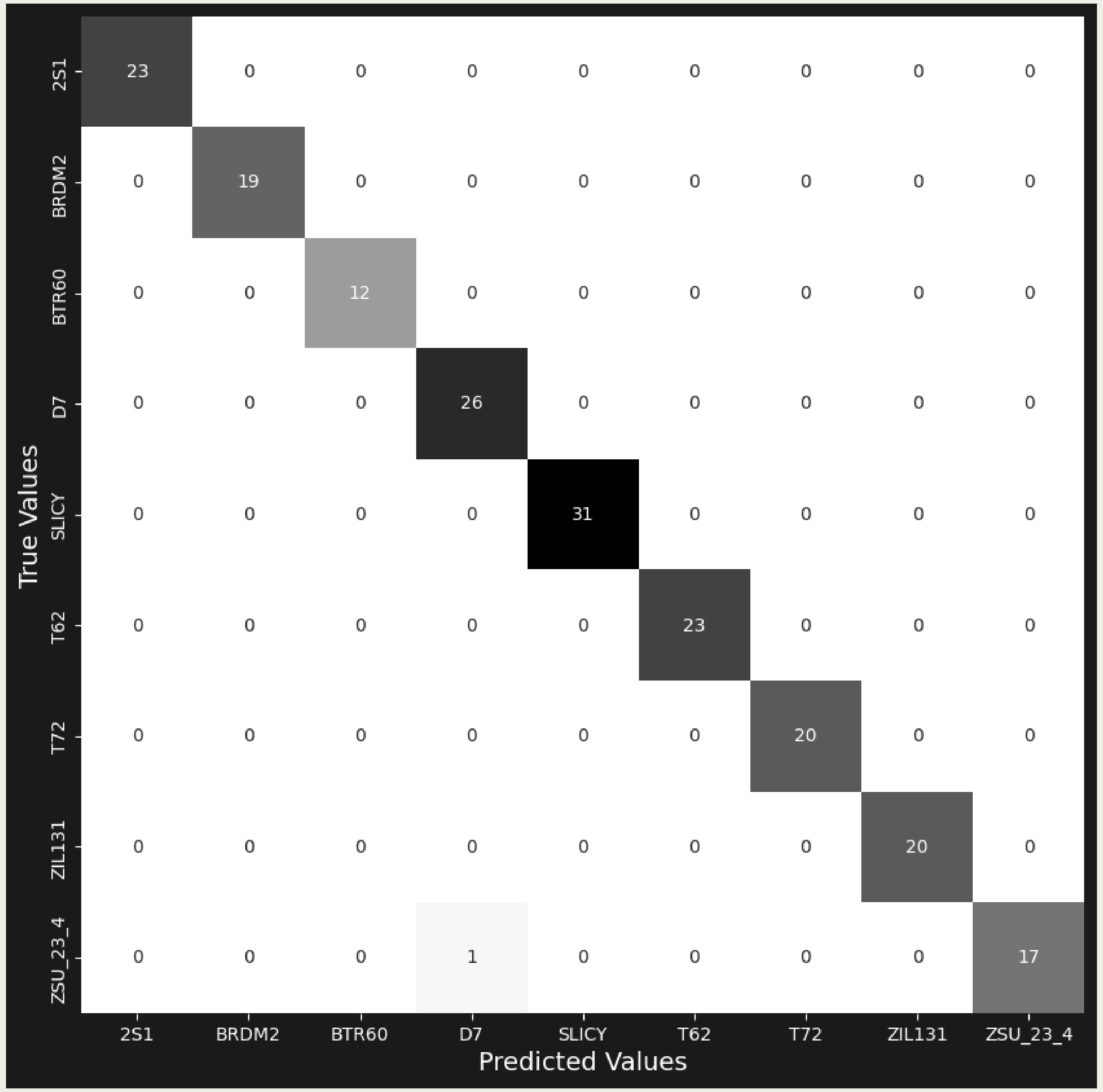


Confusion Matrix for Testing dataset

MODEL 2

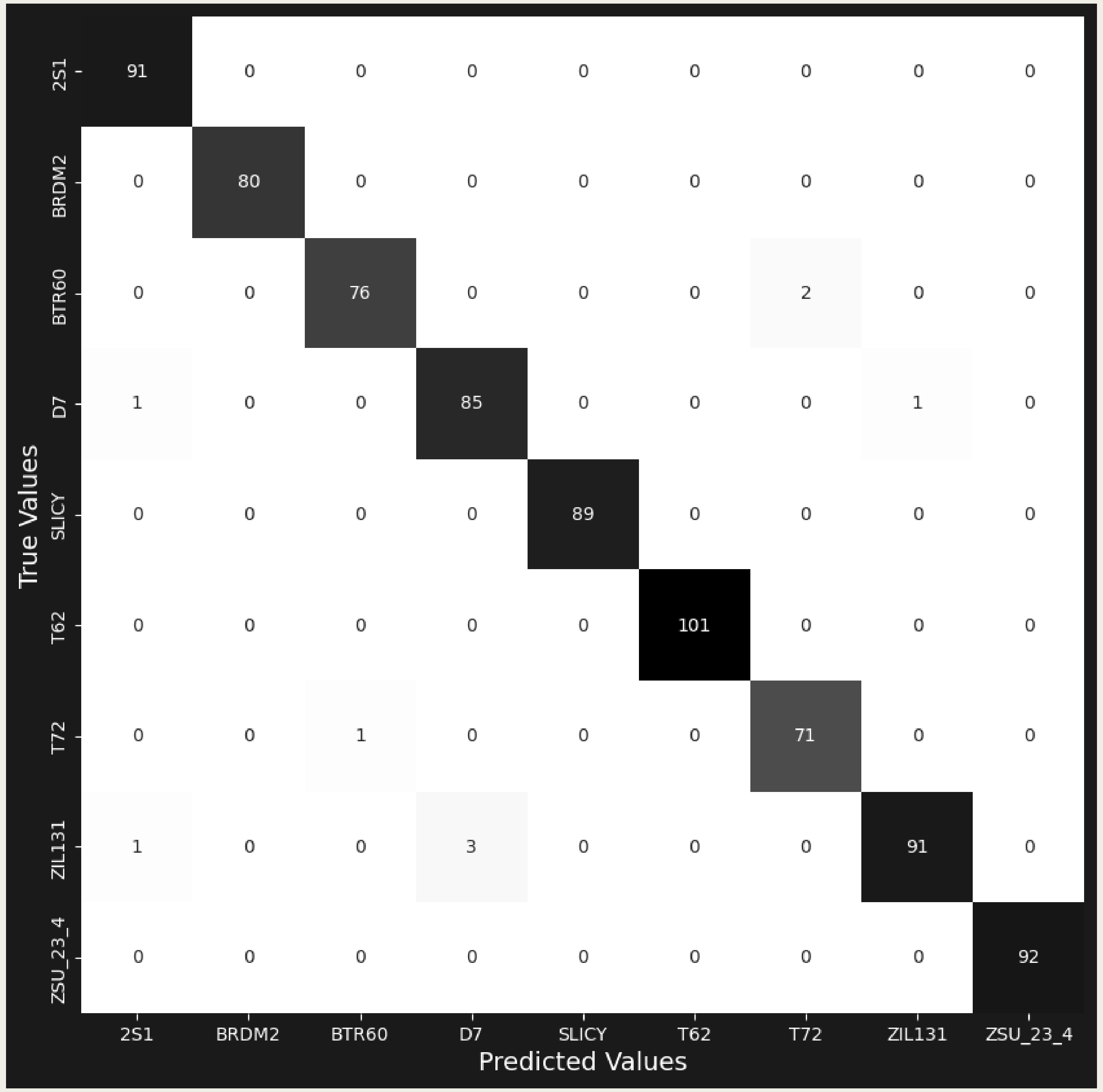


Confusion Matrix for Validation dataset

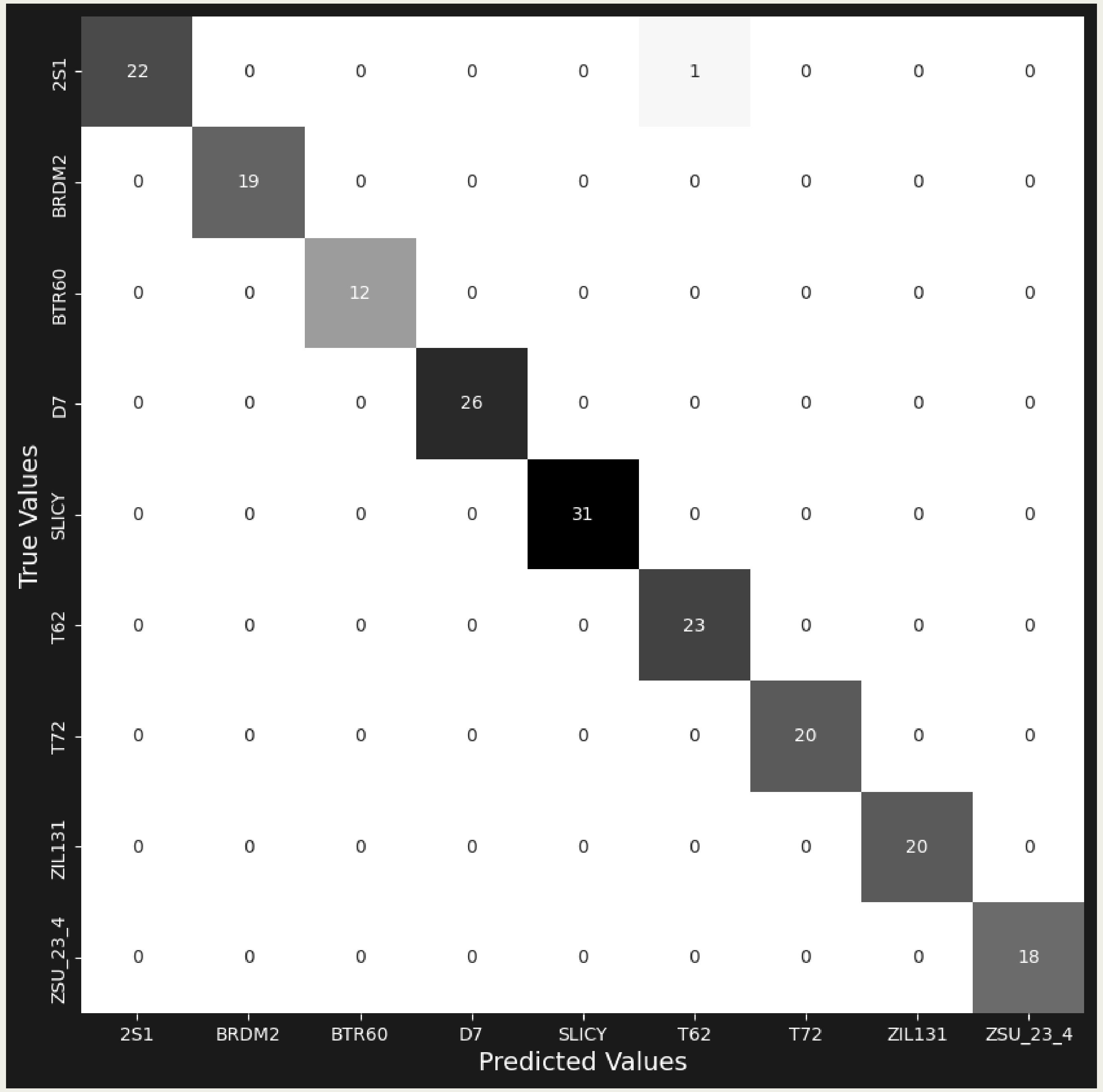


Confusion Matrix for Testing dataset

MODEL 3



Confusion Matrix for Validation dataset



Confusion Matrix for Testing dataset

Thank you!

TEAM 4