Draft version September 10, 2025
Typeset using LaTeX **twocolumn** style in AASTeX631

Machine Learning Architecture for Brightest Cluster Galaxy Candidate Classification: A Probabilistic Approach with DES Prior Integration

Documentation of Implementation[1]

[1] *High Energy Physics Image Marker Project*

ABSTRACT

We present a machine learning framework for identifying Brightest Cluster Galaxies (BCGs) that combines optical imaging data with auxiliary astronomical measurements through neural network architectures providing both deterministic classifications and probabilistic outputs. Our approach employs a two-stage pipeline: (1) candidate identification via either automatic local maxima detection or DES photometric prior catalogs, and (2) feature-based classification using multi-layer perceptrons with optional uncertainty quantification through Monte Carlo dropout and temperature scaling. The system processes BCG datasets at 2.2 and 3.8 arcminute scales, incorporating redshift and stellar mass indicators (Δm^*_z) as additional features. We implement supervised training using RedMapper BCG probabilities for loss weighting while excluding these probabilities from inference features to prevent data leakage. This framework addresses key challenges in automated BCG identification by providing calibrated confidence estimates essential for large-scale astronomical surveys.

Keywords: methods: data analysis — galaxies: clusters: general — techniques: image processing — methods: statistical

1. INTRODUCTION

The identification of Brightest Cluster Galaxies (BCGs) represents a fundamental challenge in extragalactic astronomy, requiring integration of photometric, morphological, and contextual information. Traditional approaches relying on manual inspection or simple brightness-based selection become computationally prohibitive for modern large-scale surveys such as the Dark Energy Survey (DES).

We present a machine learning framework that reformulates BCG identification as a candidate classification problem, providing both deterministic rankings and probabilistic outputs with uncertainty quantification. Our approach naturally handles the discrete nature of galaxy identification while incorporating diverse astronomical data products.

2. SYSTEM ARCHITECTURE

Figure **??** presents the complete machine learning pipeline, illustrating data flow from optical inputs through candidate selection, feature extraction, neural network processing, and probabilistic inference.

3. METHODOLOGY

3.1. *Problem Formulation*

Rather than treating BCG identification as coordinate regression, we formulate it as a candidate ranking and classification task. Given an optical image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ and optional auxiliary measurements $\mathbf{a} \in \mathbb{R}^{D_a}$ (redshift, stellar mass indicators), our objective is to:

1. Identify candidate locations $\mathcal{C} = \{(x_i, y_i)\}_{i=1}^N$ via automatic detection or prior catalogs
2. Extract feature representations $\mathbf{f}_i \in \mathbb{R}^D$ for each candidate
3. Classify each candidate with score s_i (deterministic) or probability $p_i = P(\text{BCG}|\mathbf{f}_i)$ (probabilistic)
4. Select the highest-scoring candidate as the BCG prediction

This formulation enables direct uncertainty quantification, natural handling of ambiguous cases, and principled integration of diverse observational constraints.

3.2. *Data Integration*

Our framework processes optical imaging data from BCG datasets at two angular scales:

3.2.1. *BCG Datasets*

The system supports BCG images at two scales:

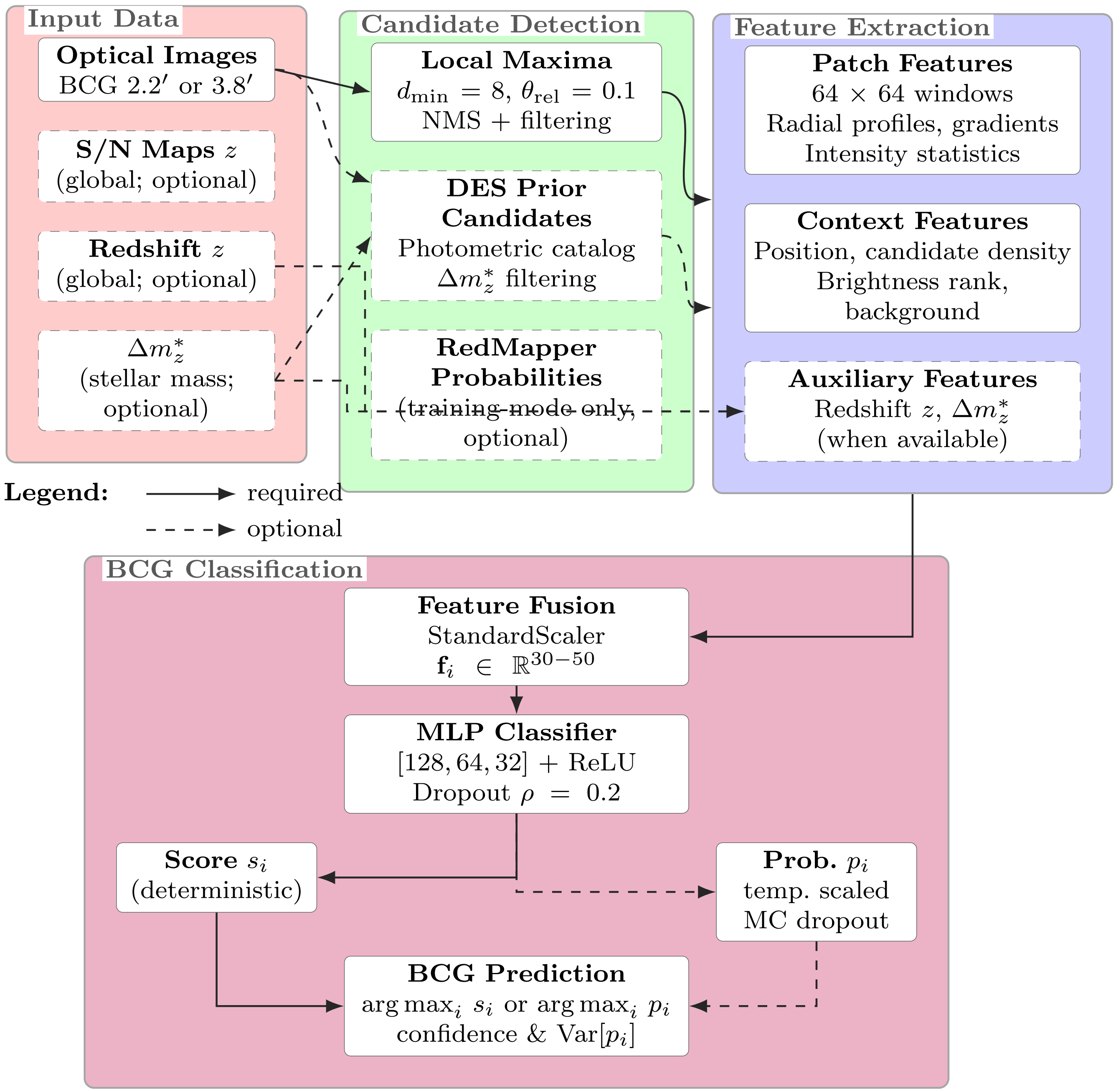

Figure 1. BCG candidate classification architecture. The system processes optical astronomical data from BCG datasets through candidate selection (automatic local maxima detection or DES photometric priors), comprehensive feature extraction, neural network classification with optional uncertainty quantification, and probabilistic inference. Solid arrows indicate required components; dashed arrows show optional enhancements available based on data configuration. The framework supports both 2.2' and 3.8' scale BCG datasets with optional auxiliary features (redshift, stellar mass indicators).

- **2.2 arcminute scale**: Higher resolution images suitable for nearby clusters
- **3.8 arcminute scale**: Wider field images accommodating more distant systems

Images are stored as multi-frame TIFF files with embedded WCS (World Coordinate System) information for astrometric calibration.

3.2.2. *Auxiliary Astronomical Measurements*

The system incorporates auxiliary measurements as global features:

- **Photometric Redshift**: z provides distance and evolutionary context

- **Stellar Mass Indicator**: Δm_z^* represents magnitude difference from characteristic stellar mass, providing crucial constraints on galaxy properties

3.3. *Candidate Selection Strategies*

The framework supports two distinct candidate identification approaches:

3.3.1. *Automatic Candidate Detection*

The automatic detection system identifies local intensity maxima using a traditional approach:

1. **Local Maxima Identification**: Apply maximum filter with 3×3 kernel
2. **Intensity Thresholding**: Apply relative threshold $\theta_{\text{rel}} = 0.1$ (default)
3. **Border Exclusion**: Remove candidates within border pixels of image boundaries
4. **Non-Maximum Suppression**: Enforce minimum separation $d_{\min} = 8$ pixels between candidates

3.3.2. *DES Photometric Prior Candidates*

For datasets with existing photometric catalogs, we utilize DES photometric priors:

- Pre-selected candidates from DES photometric pipelines
- Stellar mass-based filtering using Δm_z^* criteria
- Reduced computational overhead compared to exhaustive detection
- Integration with existing astronomical data products

3.4. *Feature Extraction*

For each candidate location (x_i, y_i), we extract comprehensive feature vectors combining local morphological information with global contextual constraints.

3.4.1. *Patch-Based Features*

Fixed-size square patches $\mathbf{P}_i \in \mathbb{R}^{64\times64\times C}$ are extracted around each candidate, computing intensity statistics, geometric moments, radial profiles, and gradient features.

3.4.2. *Contextual Features*

Beyond local morphology, we extract global contextual information including spatial context, candidate density, brightness ranking, and background statistics.

3.4.3. *Auxiliary Feature Integration*

When auxiliary measurements are available, they are concatenated to form complete feature vectors:

$$\mathbf{f}_i = [\mathbf{f}_i^{\text{patch}}, \mathbf{f}_i^{\text{context}}, \mathbf{a}] \tag{1}$$

where $\mathbf{a}$ includes redshift z and stellar mass indicator Δm_z^*.

3.5. *Neural Network Architectures*

Our framework implements two complementary network architectures:

3.5.1. *Deterministic Classifier*

The base `BCGCandidateClassifier` provides deterministic candidate rankings with hidden dimensions $[128, 64, 32]$ and dropout rate $\rho = 0.2$.

3.5.2. *Probabilistic Classifier with Uncertainty Quantification*

The `BCGProbabilisticClassifier` extends the base architecture for uncertainty-aware applications using temperature scaling and Monte Carlo dropout for epistemic uncertainty quantification.

3.6. *Training Procedures*

For each training image, we identify the candidate j^* closest to ground truth coordinates:

$$j^* = \arg\min_j \|(x_j, y_j) - (x_{\text{true}}, y_{\text{true}})\|_2 \tag{2}$$

When available, RedMapper BCG probabilities inform training through weighted loss functions while being explicitly excluded from inference features to prevent data leakage.

4. IMPLEMENTATION DETAILS

The system is implemented in Python using PyTorch for neural network components. Key modules include:

- `data.data_read_bcgs`: BCG dataset loading and coordinate processing
- `data.candidate_dataset_bcgs`: Candidate-based dataset generation for BCG data
- `utils.candidate_based_bcg`: Candidate detection and feature extraction
- `ml_models.candidate_classifier`: Deterministic classifier implementation
- `ml_models.uq_classifier`: Probabilistic classifier with uncertainty quantification
- `utils.test_desprior_candidates`: DES prior candidate integration

5. CONCLUSIONS

We have presented a machine learning framework for BCG identification that addresses key limitations through probabilistic inference and flexible candidate selection. The candidate-based formulation, combined with auxiliary feature integration and optional uncertainty quantification, provides a robust solution for automated BCG detection suitable for large-scale survey applications.

ERROR: In AASTeX v6.3.1 the `\acknowledgments` command has been deprecated.
Instead, please use the begin/end form:
`\begin{acknowledgments}...\end{acknowledgments}`
when using acknowledgments. For more details, see:
https://journals.aas.org/aastexguide/#acknowledgments