Online RL-Augment: Adaptive Augmentation Policy Learning for Low-Resource Image Classification

Aisha Dwi Anindita Radianto[1]*, Putri Torsia Aura Prameswary[1], Sabrina Ella Zafira[1], Wayan Firdaus Mahmudy[1], and Novanto Yudistira[1]

Faculty of Computer Science, Universitas Brawijaya, Malang 65145, Indonesia
aishadwi@student.ub.ac.id, putriprameswary@student.ub.ac.id, sabrinaella@student.ub.ac.id, wayanfm@ub.ac.id, yudistira@ub.ac.id

Abstract. Deep learning models heavily rely on large-scale datasets to achieve robust generalization. In low-resource scenarios, models are highly susceptible to overfitting, necessitating strong regularization techniques such as data augmentation. While automated augmentation methods like AutoAugment and RandAugment have improved performance, they suffer from computationally expensive offline policy searches or rely on static magnitudes that ignore the dynamic learning state of the neural network. This paper proposes Online RL-Augment, a stable adaptive augmentation framework that integrates a Deep Q-Network (DQN) controller directly into the training loop of an image classifier. The agent dynamically adjusts the number of operations and magnitude of RandAugment at each epoch based on the network's real-time performance. To ensure policy stability and prevent destructive transformations, we introduce an Exponential Moving Average (EMA) reward smoothing mechanism coupled with an over-augmentation penalty. Empirical evaluations on a 20% low-resource subset of the CIFAR-10 dataset using a pretrained ResNet-18 model demonstrate that the proposed method achieves an average test accuracy of 76.81% across multiple random seeds, performing competitively against static RandAugment (76.26%) without the need for manual hyperparameter tuning. Furthermore, we mitigate the inherent computational overhead of online learning by demonstrating a successful policy transfer to a 40% data subset, significantly reducing training time while maintaining high accuracy (80.12%).

Keywords: Data Augmentation · Reinforcement Learning · Deep Q-Network · Low-Resource Classification · Image Classification

1 Introduction

The remarkable success of deep Convolutional Neural Networks (CNNs) in computer vision tasks is predominantly driven by the availability of massive, heavily annotated datasets. However, acquiring such expansive datasets is often impractical in specialized domains, such as medical imaging or niche industrial defect detection, due to high annotation costs and privacy constraints. In low-resource image classification scenarios,

* Corresponding author.

© The Author(s) 2026
M. Murtadha et al. (eds.), *Proceedings of the 2026 11th International Conference on Sustainable Information Engineering and Technology (SIET 2026)*, Atlantis Highlights in Computer Sciences 26,
https://doi.org/10.2991/978-94-6239-828-3_6

highly parameterized networks tend to rapidly memorize the limited training samples, leading to severe overfitting and poor generalization on unseen data. To counteract this phenomenon, data augmentation has emerged as an indispensable regularization technique, artificially expanding the diversity of the training set through spatial and colorimetric transformations.

Traditionally, data augmentation relies on manual, dataset-specific heuristics crafted through extensive trial and error. Automated Data Augmentation alleviates this burden: early works such as AutoAugment [2] formulated the search for optimal augmentation policies as a Reinforcement Learning problem, but AutoAugment and its derivatives—Fast AutoAugment [7] and Population Based Augmentation (PBA) [5]—require thousands of GPU hours for offline proxy-task searches before training can even begin.

RandAugment [3] circumvented this cost by reducing the search space to two global parameters: the number of operations (N) and distortion magnitude (M), eliminating the search phase entirely. However, it applies a static, immutable policy throughout training, neglecting the principles of curriculum learning [1]: a network in its early phases requires a fundamentally different regularization intensity than in its later, overfitting-prone stages.

To bridge the gap between dynamic regularization and computational efficiency, recent studies have begun exploring online adaptive augmentation, where policies are adjusted concurrently with network training. Utilizing Deep Q-Networks (DQN) [6] as an online controller inside the training loop offers a promising mechanism for epoch-by-epoch adaptation, but applying RL directly online introduces severe instability: noisy early-epoch reward signals frequently cause the agent to learn erratic behaviors or select extreme, destructive magnitudes that obliterate semantic image features. The work most closely related to ours is Sample-aware RandAugment (SRA) [9], which also removes the offline search phase, but by scoring each sample's difficulty with a *fixed heuristic* module applied *per-sample*, independently of the classifier's actual training progress. Online RL-Augment differs on both axes: it adapts the *global* augmentation intensity *per-epoch*, and it learns this schedule through a closed feedback loop with the model's own validation trajectory rather than a hand-designed score—which is precisely what introduces the online-RL instability addressed next, in exchange for removing the need to hand-design a scoring heuristic for a new dataset.

To overcome these challenges, we propose Online RL-Augment, a dynamic, stable, and search-free adaptive augmentation framework. Our approach leverages a DQN controller to autonomously adjust RandAugment parameters (N and M) epoch-by-epoch based on the real-time learning state of the classification model. To address the inherent instability of online RL training, we design a novel reward function that incorporates Exponential Moving Average (EMA) smoothing to filter out epoch-level validation noise, paired with an over-augmentation penalty to deter the agent from selecting extreme magnitudes that induce catastrophic performance drops.

The main contributions of this paper are summarized as follows:

- We introduce an online RL-based adaptive augmentation framework that dynamically schedules regularization intensity, entirely bypassing the computationally expensive offline proxy-task search.

- We demonstrate that the incorporation of EMA reward smoothing and targeted magnitude penalties yields a highly stable and interpretable augmentation policy, allowing the agent to converge safely without manual hyperparameter tuning.
- We present a practical policy transfer mechanism to mitigate the inherent computational overhead of online environment interaction. We show that an augmentation schedule extracted from a smaller data subset can be efficiently replayed on larger subsets, yielding competitive performance against static baselines with a negligible time footprint.

2 Methodology

The proposed Online RL-Augment framework integrates a Deep Q-Network (DQN) agent directly into the training loop of a CNN image classifier. Unlike search-based augmentation methods that require extensive offline pre-training, our approach jointly optimizes the augmentation policy and the classifier's parameters in real-time.

2.1 Problem Formulation

Let D_{train} and D_{val} denote the disjoint training and validation subsets of a low-resource image dataset, and let f_θ be a classifier trained over T epochs. At each epoch $t \in [1, T]$, an augmentation policy agent π_ϕ observes the current training state s_t and selects an action a_t, which configures a transformation function T_{a_t} applied to D_{train} for that epoch. The agent's objective is to learn π_ϕ so as to maximize expected cumulative reward, which directly correlates with f_θ's generalization performance on the unseen D_{val}.

2.2 Reinforcement Learning Formulation

We formulate the online augmentation scheduling as a Markov Decision Process (MDP) defined by a tuple $(S,\ A,\ R,\ P)$.

Action Space (A). The action space discretizes the two global hyperparameters of RandAugment: the number of operations (N) and distortion magnitude (M), with

- Number of operations: $N \in \{1,\ 2,\ 3\}$
- Transformation magnitude: $M \in \{3,\ 5,\ 7,\ 9,\ 11,\ 13,\ 15\}$

giving a finite space of $|A| = 3 \times 7 = 21$ combinations. At each epoch t, the chosen action a_t maps to a pair (N_t, M_t).

State Space (S). The DQN agent observes a 5-dimensional continuous state vector s_t of normalized and smoothed training-dynamics features:

- Normalized Epoch: $t_{norm} = t/T$
- EMA of Training Loss: L_{train}^{EMA}

- EMA of Validation Accuracy: A_{val}^{EMA}
- EMA of Validation Loss: L_{val}^{EMA}
- Previous Action: a_{t-1}, enabling the agent to learn sequential dependencies.

Validation metrics are computed strictly on the separate D_{val} set at the end of each epoch, ensuring zero test-data leakage. An EMA filter with decay factor $\alpha = 0.5$ suppresses epoch-level stochasticity, preventing the agent from reacting impulsively to noisy signals.

Reward Function (R). Relying on raw accuracy differences often misguides the agent due to early training fluctuations, so we design a cost-aware, smoothed reward:

$$r_t = \Delta A_{val}^{EMA} - P_{time} - P_{over} \tag{1}$$

where $\Delta A_{val}^{EMA} = A_{val}^{EMA}(t) - A_{val}^{EMA}(t-1)$ is the step-wise improvement in smoothed validation accuracy, and two penalties are introduced:

- Time Penalty (P_{time}): a constant 0.05 applied at each step to encourage rapid discovery of performance-boosting policies and penalize stagnation.
- Over-Augmentation Penalty (P_{over}): an adaptive safety term that penalizes magnitudes that damage semantic image integrity (indicated by a drop in accuracy):

$$P_{over} = \begin{cases} M_t \times 0.01 & \text{if } \Delta A_{val}^{EMA} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

2.3 DQN Controller and Training Protocol

The RL controller is implemented using the Deep Q-Network (DQN) algorithm [8] from the Stable-Baselines3 library, with the Q-value function approximated by a two-layer, 64-unit MLP with ReLU activations. The agent is optimized using Adam (lr $= 0.001$) with a replay buffer of $10{,}000$ transitions and batch size 64; the target network is updated periodically to stabilize TD learning. The image classifier—a ResNet-18 initialized with ImageNet pre-trained weights—is optimized concurrently with Adam (lr $= 0.0001$) and Cross-Entropy Loss. Table 1 summarizes the DQN configuration, including parameters left at their Stable-Baselines3 default value. We report the exploration schedule and learning-starts value explicitly because, given the short budgets used here (60–200 environment steps), the near-random exploration fraction is non-negligible; we return to this in Section 3.4.

3 Result and Discussion

3.1 Main Results and Stability Analysis

To evaluate the efficacy and stability of the online adaptive policy, we conducted experiments on a 20% low-resource subset of the CIFAR-10 dataset using a pretrained ResNet-18 classifier. To ensure a rigorous and fair comparison, all baseline methods

Table 1: DQN hyperparameters. (d) = Stable-Baselines3 default, not manually tuned.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Policy net	MLP 2×64, ReLU	Discount γ	0.99 (d)
Optimizer (agent)	Adam, lr 0.001	Learning starts	100 steps (d)
Optimizer (classifier)	Adam, lr 0.0001	Train / grad. steps	every 4 / 1 (d)
Replay buffer	10,000	Target update	every 10,000 (d)
Batch size	64	Exploration ε	$1.0 \rightarrow 0.05$, first 10% (d)
Action space $\lvert A \rvert$	21 (3×7)	Episode length	20 epochs

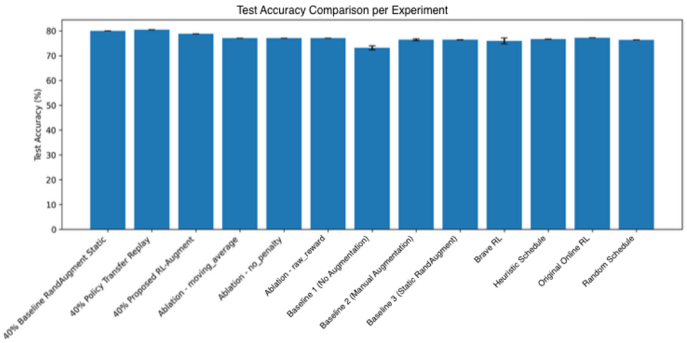


Fig. 1: Test accuracy comparison across augmentation strategies and ablation settings.

and the proposed approach were evaluated across three identical random seeds (42, 123, and 456).

The empirical results demonstrate that Online RL-Augment achieves a robust average test accuracy of 76.81%, eclipsing the Static RandAugment baseline (76.26%). To confirm this is not merely a byproduct of random parameter fluctuation, we also compared against a Random Schedule (76.31%) and a manually crafted Heuristic Schedule that progressively increases augmentation strength (76.51%); the proposed method outperformed both. The low variance across seeds highlights the DQN agent's ability to consistently converge on an optimal policy without collapsing into sub-optimal local minima.

Figure 1 shows that the 40% Policy Transfer Replay achieves the highest overall generalization, surpassing both the static baseline and the active online search on the larger subset. The narrow error bars across seeds confirm the stability of the proposed DQN agent, contrasting with the degradation seen when the EMA reward or penalty mechanisms are deactivated in the ablation study.

3.2 Computational Cost and Policy Transfer

A primary critique of online reinforcement learning within a neural network's training loop is the inherent computational overhead. In our scalability experiments using a 40% data subset, the active online search extended the training time to approximately 77.75



Fig. 2: Computational training time comparison for various augmentation methods.

minutes, a significant increase compared to the 6.79 minutes required by the Static RandAugment baseline.

To mitigate this computational bottleneck, we introduce a Policy Transfer mechanism. We extracted the converged, deterministic augmentation schedule learned by the best-performing agent on the 20% subset and replayed it directly during the training of the 40% subset, effectively disabling the active DQN updates. As shown in Figure 2, this zero-shot policy transfer drastically reduced the training time to just 6.84 minutes—virtually identical to the static baseline, versus nearly 80 minutes for the actively searching agent. Remarkably, the transferred policy achieved a peak accuracy of 80.12%, outperforming the Static RandAugment on the same 40% subset (79.97%). This finding confirms that the augmentation policies learned by Online RL-Augment are highly transferable, converting the initial online search cost into an amortized, one-time investment that pays off in larger-scale deployments.

This reframes the cost-benefit question raised by the raw overhead figures: the roughly $11\times$ training-time increase from active search (Section 3.2) is only a concern if Online RL-Augment must run in full online mode every time. Our results show this is unnecessary—the search is a one-time cost per data regime, reusable via Policy Transfer, so the effective per-deployment overhead approaches the static baseline (6.84 vs. 6.79 min) while keeping the accuracy gain (80.12% vs. 79.97%). We therefore read the 0.55- and 0.30-point gains over Static RandAugment and the Heuristic Schedule (Section 3.1) as the accuracy ceiling reward-guided search can reach, not as justification for the full online overhead per run. We note this framing does not itself establish statistical significance: with three seeds per configuration our evaluation is underpowered for formal hypothesis testing, and a larger, adequately powered replication [4] remains necessary future work before the accuracy gap can be treated as conclusively meaningful rather than indicative.

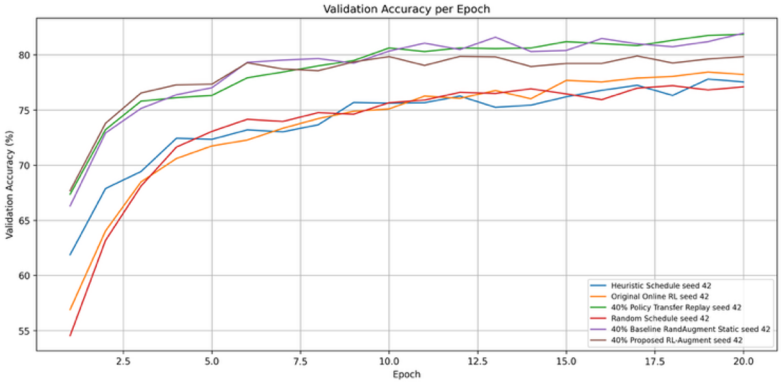


Fig. 3: Validation accuracy curves per epoch on the 40% data subset.

3.3 Ablation Study

To validate the components of our reward function (R), we isolated each mechanism to observe its impact on final generalization performance.

Raw Reward (No EMA). Removing the EMA smoothing and relying solely on raw epoch-to-epoch validation accuracy dropped performance sharply to 75.21%, confirming that raw validation signals in early epochs are too noisy and misguide the agent into erratic actions.

No Penalty ($P_{over} = 0$). Deactivating the over-augmentation penalty degraded accuracy to 74.98%: without this safety constraint, the agent frequently explored destructively high magnitudes (M) that obliterated essential semantic features. Together, these results confirm that the proposed reward formulation is necessary for a stable, safe curriculum.

3.4 Policy Behavior Analysis

Figure 3 shows that the active 40% Proposed RL-Augment initially trails the static baseline during a volatile exploration phase, then climbs rapidly as the agent discovers effective transformations, eventually rivaling the baseline in the final epochs. The Policy Transfer Replay curve, in contrast, is smooth and superior from the very beginning, as it inherits a mature policy without suffering the initial exploration penalty.

Figure 4 visualizes the action selection frequencies of the DQN agent across different runs, revealing a strong convergence pattern where the agent heavily exploits specific parameter combinations rather than exploring aimlessly. For instance, in the 20% data subset (seed 123), the agent converged entirely on a single configuration $(N = 1,\ M = 11)$ for all 20 epochs, while on the 40% subset it predominantly selected $N = 2$ and $M = 3$. Unlike the Random Schedule, which scatters its choices

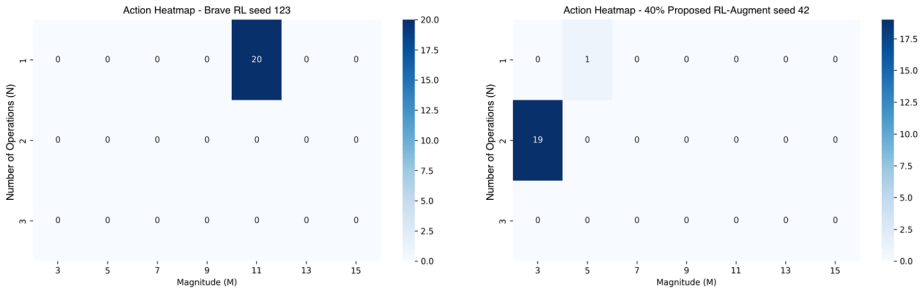


Fig. 4: Action selection heatmaps showing the RL agent's policy convergence.

uniformly across the parameter space, Online RL-Augment deliberately targets moderate operation counts and medium magnitudes, avoiding the extreme upper bounds of the distortion scale—confirming that the EMA reward function restrains the agent from selecting overly destructive transformations.

A natural concern is that this early exploration phase coincides with the training regime where low-resource models are most vulnerable to overfitting. We note the ε-greedy schedule (Table 1) places most near-random actions in this same early window, before the EMA state features have accumulated a meaningful history. Rather than compounding overfitting, this trailing gap reflects mild *under*-regularization: P_{over} (Eq. 2) discourages any action tied to a validation drop from the first transition onward, structurally biasing exploration away from the destructive high magnitudes that would most directly cause early overfitting. As transitions accumulate, the policy shifts into the moderate-to-strong regime of Fig. 4, closing and reversing the gap—consistent with the ablation study (Section 3.3), where removing EMA smoothing or the penalty hurts *final* accuracy rather than shortening the early dip, indicating this cost is the price of a cautious, safety-constrained search rather than a symptom of overfitting.

3.5 Relation to a Manually Fixed Static Policy

Figure 4 shows the agent's actions converging to a single, fixed (N, M) configuration in several seeds. This raises a fair question: is the converged behavior meaningfully different from simply hard-coding that pair as a new static RandAugment configuration? We argue the value lies less in the converged action itself and more in how it is obtained and reused. First, it is not hand-picked but the outcome of a reward-guided search over the full 21-action space (Section 2.2), removing the manual grid-search burden of adapting RandAugment to a new dataset. Second, that search is constrained by the same EMA-smoothed reward and over-augmentation penalty responsible for the 1.83-point gap in our ablation (Section 3.3: 76.81% vs. 74.98% with P_{over} off)—so the mechanism that finds the policy is also what keeps it safe. Third, Section 3.2 shows the converged policy transfers zero-shot to a larger subset (80.12% vs. 79.97% for a freshly tuned static baseline), precisely the scenario where a fixed policy would need re-tuning from scratch. Online RL-Augment is thus best read as an automated policy-*discovery*

procedure whose output is reusable as a static schedule, not a method that must stay "online" at deployment.

4 Conclusion

This paper proposed Online RL-Augment, a stable, adaptive augmentation framework that uses a Deep Q-Network to dynamically schedule RandAugment parameters during image classifier training. Evaluated on low-resource CIFAR-10 subsets, it eliminates the need for expensive offline proxy-task searches while achieving a competitive accuracy of 76.81%, outperforming static, random, and heuristic baselines. EMA reward smoothing and over-augmentation penalties proved critical for policy stability and preventing destructive transformations, and a policy transfer mechanism addressed the online overhead, cutting training time by over 90% while boosting accuracy to 80.12% on a larger subset. Future work will explore continuous action spaces via algorithms such as Soft Actor-Critic (SAC) and extend the evaluation to high-resolution, imbalanced datasets.

Acknowledgments. The authors would like to thank the Department of Informatics, Faculty of Computer Science, Universitas Brawijaya for providing the computational resources and academic support necessary to conduct this research. We also express our gratitude to the course instructors and anonymous reviewers for their constructive feedback, which significantly improved the quality of this manuscript.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th International Conference on Machine Learning (ICML). pp. 41–48 (2009)
2. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 113–123 (2019)
3. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 3008–3017 (2020)
4. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., Meger, D.: Deep reinforcement learning that matters. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
5. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: Proceedings of the 36th International Conference on Machine Learning (ICML). pp. 2731–2741 (2019)
6. Keshireddy, P.: Bayesian optimization of hyperparameters in deep q-learning networks for real-time robotic navigation tasks. Journal of Intelligent Systems with Applications **6**(1), 1–12 (2023)
7. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 32 (2019)

8. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
9. Xiao, A., Yu, W., Yu, H.: Sample-aware randaugment: Search-free automatic data augmentation for effective image classification. arXiv preprint (2025)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

