



Computer Vision

Multi-Label Image Classification
Query2Label & MLSPL

Group 25

Riajul Islam, Andreas C. Kreth, Christine Midtgaard

June 19, 2025

Aarhus University

- Problem & Motivation
- Related works
- Background concepts
- Method 1: Query2Label (Q2L)
- Method 2: Multi-label Learning from Single Positive Labels (MLSPL)
- Experiments
- Results & Discussion
- Conclusion

Problem & Motivation

What is Multi-label Classification?

- Assigns multiple labels to a single instance (e.g., an image with both “dog” and “cat”).
- Unlike single-label (one label per image) or multi-class (choose one among many), multi-label allows overlapping labels.



| | Multi-Class | Multi-Label |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| C = 3 | <p>Samples</p>  <p>Labels (t)</p> <p>[0 0 1] [1 0 0] [0 1 0]</p> | <p>Samples</p>  <p>Labels (t)</p> <p>[1 0 1] [0 1 0] [1 1 1]</p> |

Illustration of single-label vs multi-class vs multi-label classification. From <https://medium.com/@wongsirikuln/fire-alert-system-with-multi-label-classification-model-explained-by-gradcam-bc18affe178c>.

Applications

- Image tagging
- Medical imaging
- Autonomous driving
- Etc.

Challenges

- Small objects and background clutter (feature recognition).
- Severe class imbalance.
- Growing annotation cost.
- Sparse annotations.

Project goals

- Implemented two approaches:
Q2L and **MLSPL**.
- Reproduce results on MS-COCO 2014.
- Analyze resource trade-offs and reproducibility.

- Loss functions
- Locating areas of interest
- PU Learning
- Partially observed labels
- Attention and Transformer architectures

Concepts

- Multi-label Classification (MLC)
- Convolutional Neural Networks (CNNs)
- Transformers
- Loss functions

Background

Multi-label classification:

The goal is to create a model that outputs the probability $p = [p_1, \dots, p_K]$ of the presence of a category K in an input image $x \in \mathcal{X}$.

Labels $y = [y_1, \dots, y_K]$ from the label space $\mathcal{Y} = \{0, 1\}^K$.

- $y_k = 1$: k is present.
- $y_k = 0$: otherwise.










| | Multi-Class | Multi-Label |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C = 3 | | |
|    | <p>Samples</p>    <p>Labels (t)</p> <p>[0 0 1] [1 0 0] [0 1 0]</p> | <p>Samples</p>    <p>Labels (t)</p> <p>[1 0 1] [0 1 0] [1 1 1]</p> |

Illustration of single-label vs multi-class vs multi-label classification. From <https://medium.com/@wongsirikuln/fire-alert-system-with-multi-label-classification-model-explained-by-gradcam-bc18affe178c>.

CNNs: Three main types of layers:

- Convolutional layers: extract spatial features.
- Pooling layers: reduce dimensionality and summarise information.
- Fully Connected (FC) layer: interpret features for classification.

Transformers:

Vision Transformers (ViTs):

Background

Loss functions

- Measures distance between prediction (model output) and ground truth.
- Guides parameter optimization.
- Multi-label learning adds complexity: an image may trigger multiple simultaneous decisions.
- Choosing an appropriate loss is therefore important.

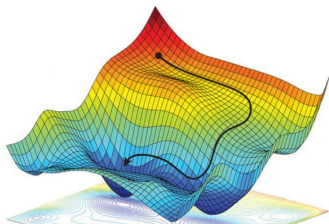


Illustration of a loss landscape.

Loss functions covered:

- Binary Cross-Entropy (BCE)
- Expected Positive Regularisation (EPR)
- Regularised Online Label Estimation (ROLE)
- Focal and Asymmetric Focal Losses

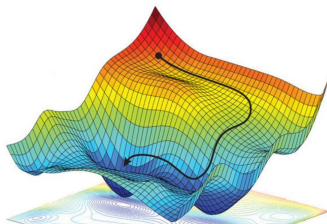


Illustration of a loss landscape.

Binary Cross-Entropy (BCE):

- Independent positive/negative prediction per class: $y_i \in \{0, 1\}$
- Model outputs probability $p_i \in [0, 1]$ for each class $i \in \{1, \dots, K\}$.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{K} \sum_{i=1}^K [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

BCE limitations:

- **Class imbalance:** gradients dominated by frequent classes \rightarrow rare classes under-represented.
- **Missing labels:** datasets rarely fully annotated. BCE treats unobserved labels as negatives, introducing false-negative bias.

Positive-only BCE used when only the observed positive label $z_{ni} = 1$ is known:

$$\mathcal{L}_{\text{BCE}}^+(\mathbf{f}_n, \mathbf{z}_n) = - \sum_{i=1}^K \mathbf{1}[z_{ni} = 1] \log f_{ni}$$

Expected Positive Regularisation (EPR)

$$\mathcal{L}_{EPR}(\mathbf{F}_B, \mathbf{Z}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{BCE}^+(\mathbf{f}_n, \mathbf{z}_n) + \lambda R_{\kappa}(\mathbf{F}_B),$$

- κ : expected positives per image (domain prior).
- $\hat{\kappa}$: average predicted positives in the batch.
- Batch penalty: $R_{\kappa}(\mathbf{F}_B) = \left(\frac{\hat{\kappa}(\mathbf{F}_B) - \kappa}{K} \right)^2$.

Regularised Online Label Estimation (ROLE)

- Jointly trains classifier $f(\cdot; \theta)$ and label-estimator $g(\cdot; \phi)$.
- Alternating updates:

$$\mathcal{L}'(\mathbf{F}_B, \tilde{\mathbf{Y}}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}_{BCE}(\mathbf{f}_n, \text{sg}(\tilde{\mathbf{y}}_n)) + \mathcal{L}_{EPR}(\mathbf{F}_B, \mathbf{Z}_B)$$

- Softly imputes missing labels ($0 < y_{ni} < 1$) and reduces false negatives.

$$\mathcal{L}_{ROLE}(\mathbf{F}_B, \tilde{\mathbf{Y}}_B) = \frac{\mathcal{L}'(\mathbf{F}_B | \tilde{\mathbf{Y}}) + \mathcal{L}'(\tilde{\mathbf{Y}} | \mathbf{F}_B)}{2}$$

Focal loss

$$\mathcal{L}_{\text{FL}} = -\frac{1}{K} \sum_{i=1}^K \alpha_i (1 - p_i)^{\gamma} y_i \log p_i$$

- Down-weights easy examples; focuses on hard, misclassified ones.
- Hyperparameters: focusing γ and class weight α .

Assymmetric focal loss

- Variant tailored for imbalance in multi-label settings.
- Uses different focusing parameters for positive vs. negative terms.

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^K \begin{cases} (1 - p_k)^{\gamma^+} \log(p_k), & \text{if } y_k = 1 \\ (p_k)^{\gamma^-} \log(1 - p_k), & \text{if } y_k = 0 \end{cases}$$

Key Take-aways

- BCE \rightarrow solid baseline but fragile under imbalance and missing labels.
- EPR and ROLE repair missing-label bias with priors and online estimation.
- Focal variants tackle severe class imbalance.
- Choice of loss crucial for high multi-label performance (see results section).

- **Feature localization:** Query2Label: A Simple Transformer Way to Multi-Label Classification (Q2L)
- **Sparse label annotation:** Multi-Label Learning from Single Positive Labels (MLSPL)

Query2Label: Core Idea

- Treat each label as a *learnable query* in a Transformer decoder.
- Decoder cross-attends to backbone feature map.
- Produces class-specific feature vectors.

- Stage 1: CNN/ViT backbone extracts spatial features.
- Stage 2: Multi-layer Transformer decoder with K queries.
- Linear head $\rightarrow K$ sigmoid outputs (one per label).
- Loss: Asymmetric Focal (AF) to mitigate imbalance.

Q2L Training Setup

- 80 epochs, AdamW, RandAugment, EMA.
- Backbones: ResNet-101, TResNet-L, Swin-L, CvT-w24.
- Memory constraints encountered on 12 GB GPUs.

MLSPL: Learning from One Positive Label

- Extreme weak supervision: exactly **one** positive per image.
- Objective: recover missing positives during training.

ROLE: Regularised Online Label Estimation

- Maintain soft estimates \hat{y} for unobserved labels.
- Jointly optimise classifier & label estimator.
- Batch-level regulariser keeps expected positives $\approx \kappa$.

Dataset: MS-COCO 2014

- 82 k train / 40 k val images, 80 object categories.
- Avg. 2.9 labels / image.
- Single-positive variant generated for MLSPL.

Experimental Setup

- Reproduced authors' pipelines under 12 GB GPU budget.
- Hyper-parameter grid: $LR \in [10^{-2}, 10^{-5}]$, $batch \in \{8, 16\}$.
- MLSPL: linear classifier then fine-tuning.

Results: Q2L (mAP)

- ResNet-101 $448^2 \rightarrow$ **84.9%**.
- TResNet-L-22k $448^2 \rightarrow$ **89.2%**.
- CvT-w24-22k $384^2 \rightarrow$ **91.3%** (SOTA).

Results: MLSPL (ROLE)

- Linear classifier → **66.3% mAP**.
- Fine-tuned ResNet-50 → **66.9% mAP** (i paper).
- Achieved with 20× fewer true labels.

- Q2L excels at localisation but is memory-hungry.
- MLSPL robust under weak labels, lightweight.
- Complementary strengths → potential hybrid model.

Limitations

- Q2L on ViTs exceeds 12 GB VRAM.
- MLSPL relies on accurate κ estimation.
- Experiments limited to COCO – needs broader validation.

Conclusions & Future Work

- Both methods reproduced successfully.
- Transformer label queries push SOTA accuracy.
- ROLE shows promise for cost-efficient annotation.
- Next steps: larger datasets, unify approaches.