Computer Vision

Multi-Label Image Classification
Query2Label & MLSPL

Group 25
Riajul Islam, Andreas C. Kreth, Christine Midtgaard

June 19, 2025

Aarhus University

Overview

- 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.

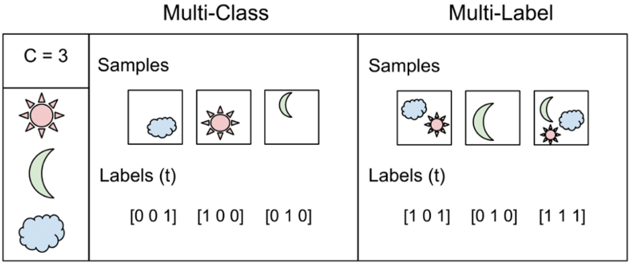


Illustration of single-label vs multi-class vs multi-label classification. From https://medium.com/@wongsirikuln/fire-alert-system-with-multi-label-classification-model-explained-by-gradcam-bc18affe178c.

Problem & Motivation

Applications

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

Problem & Motivation

Challenges

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

Problem & Motivation

Project goals

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

Related work

- 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

Multi-label classification:

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

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

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

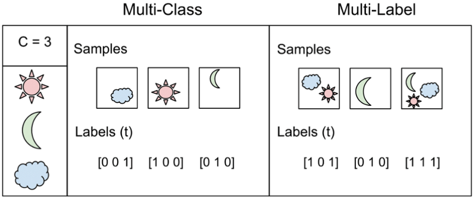


Illustration of single-label vs multi-class vs multi-label classification. From `https://medium.com/@wongsirikuln/fire-alert-system-with-multi-label-classification-model-explained-by-gradcam-bc18affe178c`.

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.

Background

Transformers:

Vision Transformers (ViTs):

Why Loss Functions Matter

- **Loss:** distance between model output and ground truth.
- Guides parameter optimization.
- Multi-label learning adds complexity: an image may trigger multiple simultaneous decisions.

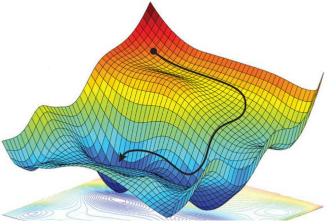


Illustration of a loss landscape.

Background

Loss functions covered:

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

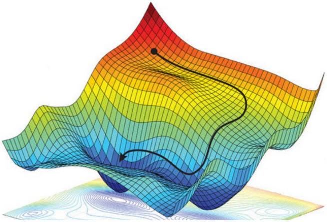


Illustration of a loss landscape.

Binary Cross–Entropy (BCE)

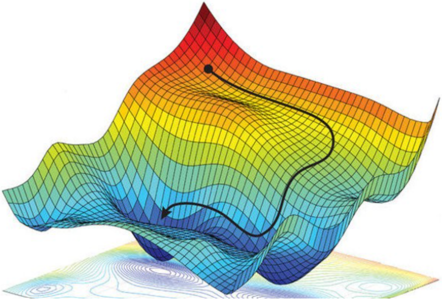
Use–case: Independent yes/no decision per class.

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

- K: number of classes; $y_i \in \{0,1\}$, $p_i \in [0,1]$.
- Widely used baseline metric.

BCE in Practice: Limitations

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



Example of sparse annotation (only one label per image).

Expected Positive Regularisation (EPR)

$$\mathcal{L}_{\mathrm{EPR}}(\mathbf{F}_B, \mathbf{Z}_B) = \frac{1}{|B|} \sum_{n \in B} \mathcal{L}^{+}_{\mathrm{BCE}}(\mathbf{f}_n, \mathbf{z}_n) + \lambda \big(\hat{\kappa}(\mathbf{F}_B) - \kappa \big)^2$$

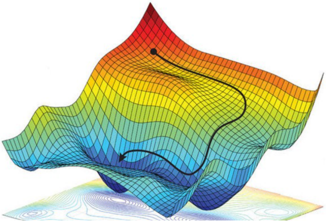
- κ: expected positives per image (domain prior).
- $\hat{\kappa}$: average predicted positives in the batch.
- Penalises deviation from prior; mitigates false–positives when only few labels are observed.

Regularised Online Label Estimation (ROLE)

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

$$\mathcal{L}'(F_B, Y_B) = \frac{1}{|B|} \sum_n \mathcal{L}_{\mathrm{BCE}}(f_n, \mathrm{sg}(y_n))$$

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



Classifier $\leftrightarrow$ Label–estimator loop.

Focal & Asymmetric Focal Losses

- **Focal Loss (FL):** down–weights easy negatives, focuses on hard examples.

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

- **Asymmetric Focal Loss (AFL):** separate focusing for positive/negative parts.

$$\mathcal{L}_{\mathrm{AFL}} = -\frac{1}{K}\sum_{i} \left[y_i (1 - p_i)^{\gamma^+} \log p_i + (1 - y_i) p_i^{\gamma^-} \log(1 - p_i) \right]$$

- Useful when negatives hugely outnumber positives.

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).

Method

- **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.

Q2L Architecture

- 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 $\rightarrow$ **66.3% mAP**.
- Fine-tuned ResNet-50 $\rightarrow$ **66.9% mAP** (¿ paper).
- Achieved with $20\times$ fewer true labels.

Discussion

- Q2L excels at localisation but is memory-hungry.
- MLSPL robust under weak labels, lightweight.
- Complementary strengths $\rightarrow$ 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.