# A CNN-RNN Framework for Image Annotation from Visual Cues and Social Network Metadata


Tobia Tesan[1], Pasquale Coscia[2], Lamberto Ballan[2]

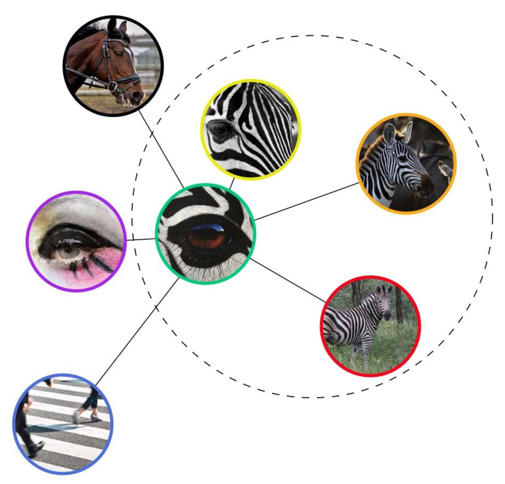
[1]Quantexa Ltd, London, UK

[2]Department of Mathematics, University of Padova, Italy


DIPARTIMENTO MATEMATICA

## 1. Image Annotation

- Image classification is a challenging task requiring additional information to correctly annotate images.
- We blend visual features extracted from neighbors and their metadata.
- Several convolutional and recurrent neural networks (CNNs-RNNs) are jointly adopted to infer similarity among neighbors and query images.



## 2. Our approach

For a query image $x$, we (nonparametrically) generate a neighborhood $\mathcal{Z}_x$ using metadata and train our neural networks to classify $x$ given $\mathcal{Z}_x$ . The set of candidate neighborhoods for an image $x$ is the set:

$$\mathcal{Z}_x = \{s \in \mathcal{P}(X) : |s| = m\}$$

The prediction $s(x, \theta)$ is the average of $f(x, \vec{z}; \theta)$ over all candidate neighbourhoods:

$$s(x, \theta) = \frac{1}{|Z_x|} \sum_{z \in \mathcal{Z}_x} f(x, \vec{z}; \theta)$$

Our models are trained minimizing the following loss function $\mathcal{L}$ :

$$\theta^* = \arg\min_{\theta} \sum_{(x,y)\in D_{train}} \mathcal{L}(s(x, \theta), y)$$

## 3. Metadata Encoding

To correctly recover similar images, our models use metadata which are directly fed to the final layers of NNs after a transformation step.

- *One-hot encoding*: sum of one-hot vectors for all relevant tags of the query image. Neighborhoods are computed using the Jaccard distance $\mathcal{J}$ between binary vectors.

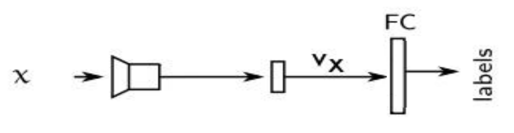
$$o_x = \sum_{i \text{ s.t. } t_i \in \{t_{(1)}, t_{(2)}, \ldots, t_{(n)}\}} e_i^{\tau}$$

- *Semantic-aware encoding*: a transformation step is employed to map binary vectors to a meaningful semantic space. *Word2vec* and *WordNet* embeddings are considered.

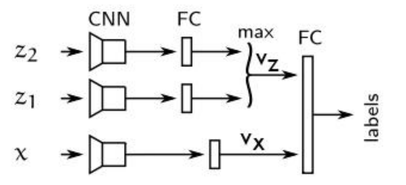
$$\rho(o_x; \beta) = \sum_{i=1}^{\tau} o_{x(i)} \cdot \beta(t_{(i)})$$

## 4. Visual Models

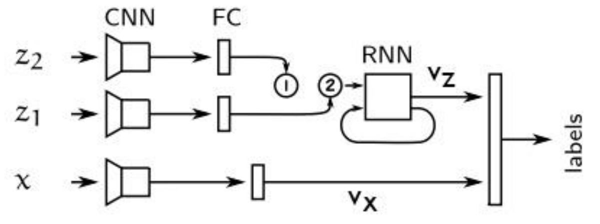
Query images are classified combining only visual features extracted by the *AlexNet* CNN pre-trained on ImageNet.




*Visual only*



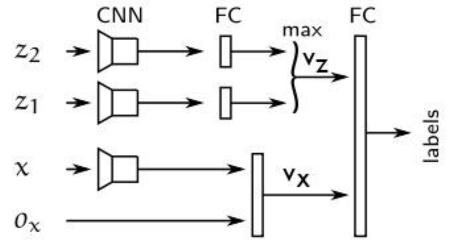

*LTN*



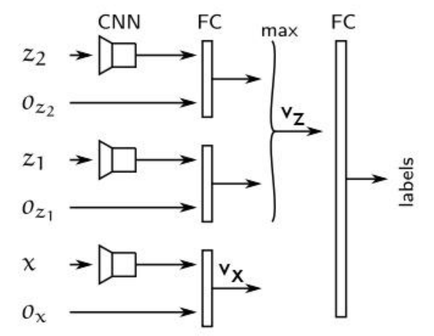

*RTN*

## 5. Joint Models

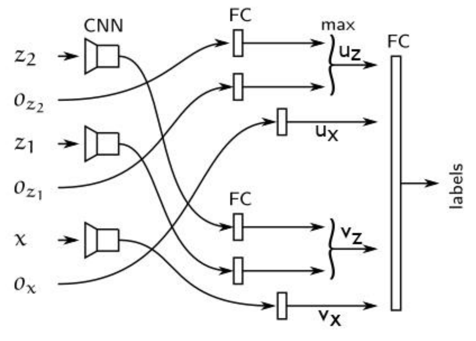
Joint models take advantage of additional information (tags) that is fed to the classification layer after a transformation step.



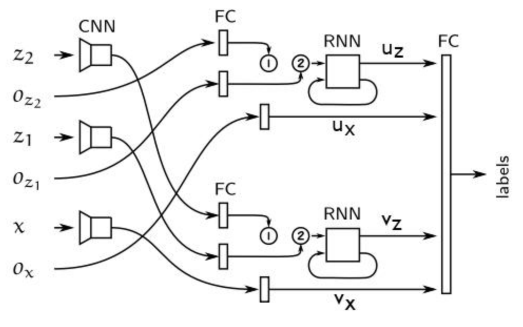
*LTN+Vecs*



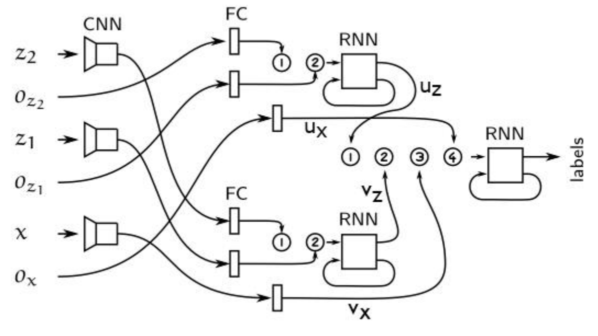
*LTN+AllVecs*



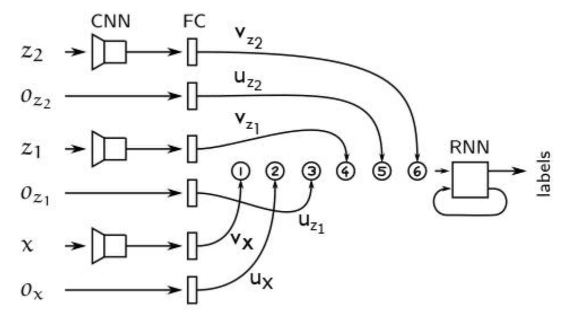
*LTwin*



*LTwin+RNN*



*LTwin+2RNN*



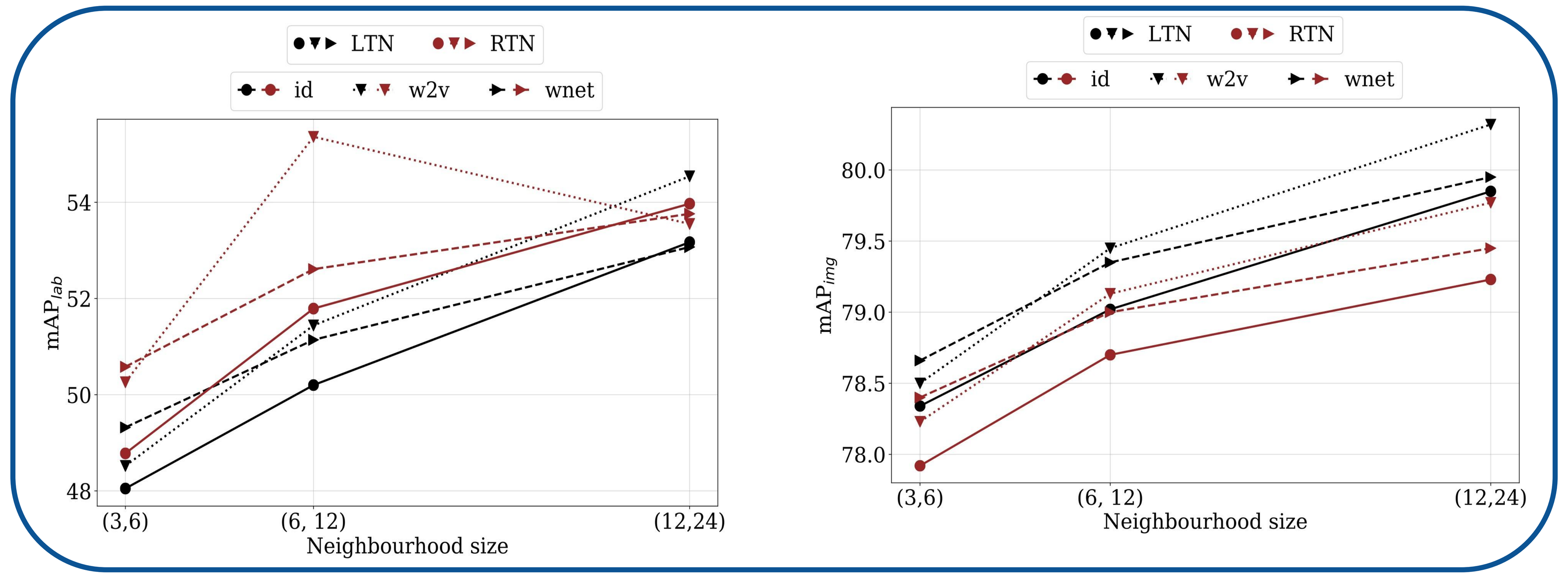
*LZip*

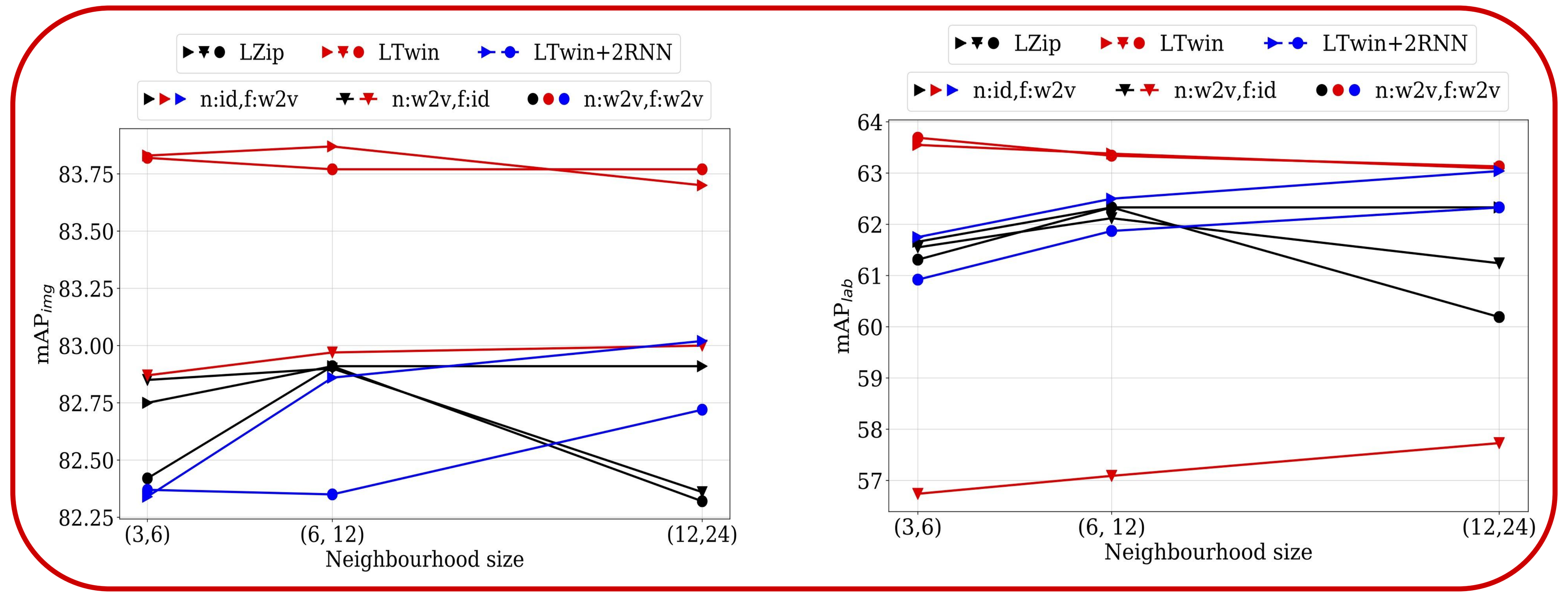
## 6. Experimental Results

Our experiments are performed on the NUS-WIDE dataset considering ~190,000 images and 5000 tags. Per-image/per-label mean Average Precisions (mAPs) metrics show that *LTwin* model achieves SOTA results compared to several baselines and all the models outperform the visual-only baseline.

| Method | $mAP_{lab}$ | $mAP_{img}$ | $rec_{lab}$ | $prec_{lab}$ | $rec_{img}$ | $prec_{img}$ |
|---|---|---|---|---|---|---|
| Tag-only Model + linear SVM [7] | 46.67 | - | - | - | - | - |
| Graphical Model (all metadata) [7] | 49.00 | - | - | - | - | - |
| CNN + WARP [16] | - | - | 35.60 | 31.65 | 60.49 | 48.59 |
| CNN-RNN [21] | - | - | 30.40 | 40.50 | 61.70 | 49.90 |
| SR-RNN [22] | - | - | 50.17 ⋆ | 55.65 ⋆ | 71.35 ⋆ | 70.57 ⋆ |
| SR-RNN + Vecs [22] † | - | - | 58.52 ⋆ | 63.51 ⋆ | 77.33 ⋆ | 76.21 ⋆ |
| SRN [35] | 60.00 | 80.60 | 41.50 ⋆ | 70.40 ⋆ | 58.70 ⋆ | 81.10 ⋆ |
| MangoNet [33] | 62.80 | 80.80 | 41.00 ⋆ | 73.90 ⋆ | 59.90 ⋆ | 80.60 ⋆ |
| LTN [2] | 52.78 ±0.34 | 80.34 ±0.07 | 43.61 ±0.47 | 46.98 ±1.01 | 74.72 ±0.16 | 53.69 ±0.13 |
| LTN + Vecs [2] † | 61.88 ±0.36 | 80.27 ±0.08 | 57.30 ±0.44 | 54.74 ±0.63 | 75.10 ±0.20 | 53.46 ±0.09 |
| Upper bound | 100.00 ±0.00 | 100.00 ±0.00 | 65.82 ±0.35 | 60.68 ±1.32 | 92.09 ±0.10 | 66.83 ±0.12 |
| Our baseline: v-only | 45.05 ±0.11 | 76.88 ±0.11 | 42.31 ±0.59 | 43.74 ±1.07 | 71.41 ±0.13 | 51.36 ±0.13 |
| Our baseline: $LTN_{n:id}$ | 53.17 ±0.12 | 79.82 ±0.16 | 45.67 ±1.75 | 47.64 ±2.18 | 74.29 ±0.13 | 53.34 ±0.17 |
| Our baseline: LTN + $Vecs_{n:id,f:id}$ † | 54.86 ±0.20 | 81.34 ±0.15 | 46.56 ±1.39 | 50.10 ±1.70 | 75.67 ±0.17 | 54.37 ±0.14 |
| Our model: $RTN_{n:w2v}$ | 55.36 ±0.34 | 79.77 ±0.27 | 48.73 ±2.77 | 51.21 ±2.61 | 74.35 ±0.29 | 53.28 ±0.24 |
| Our model: $LTwin_{n:w2v,f:w2v}$ † | **63.13** ±0.31 | **83.77** ±0.06 | 54.40 ±1.33 | 51.86 ±1.58 | 78.06 ±0.05 | 55.78 ±0.13 |

Visual Models    Joint Models





E-mail: tobiatesan@quantexa.com, {pasquale.coscia, lamberto.ballan}@unipd.it