Improving the performance of lightweight CNNs for binary classification using Quadratic Mutual Information regularization

Maria Tzelepi[a,*], Anastasios Tefas[a]

[a]*Department of Informatics*
Aristotle University of Thessaloniki
Thessaloniki, 54124, Greece

Abstract

In this paper, we propose regularized lightweight deep convolutional neural network models, capable of effectively operating in real-time on-drone for high-resolution video input. Furthermore, we study the impact of hinge loss against the cross entropy loss on the classification performance, mainly in binary classification problems. Finally, we propose a novel regularization method motivated by the Quadratic Mutual Information, in order to improve the generalization ability of the utilized models. Extensive experiments on various binary classification problems involved in autonomous systems are performed, indicating the effectiveness of the proposed models. The experimental evaluation on four datasets indicates that hinge loss is the optimal choice for binary classification problems, considering lightweight deep models. Finally, the effectiveness of the proposed regularizer in enhancing the generalization ability of the proposed models is also validated.

Keywords: Hinge Loss, Cross Entropy Loss, Binary Classification Problems, Quadratic Mutual Information, Regularizer, Lightweight Models, Real-time, Convolutional Neural Networks, Deep Learning

*Corresponding author
Email addresses: mtzelepi@csd.auth.gr (Maria Tzelepi), tefas@csd.auth.gr (Anastasios Tefas)

1. Introduction

Deep learning algorithms, [1, 2], and principally the deep Convolutional Neural Networks (CNN), [3], have been established among the most effective research directions in a wide range of computer vision tasks, [4], accomplishing outstanding performance over previous shallow models. More specifically, deep CNNs have been successfully applied in image classification [5, 6], object detection [7, 8, 9] and retrieval [10, 11, 12], visual tracking [13, 14], video captioning [15], and pose estimation [16]. Deep CNNs owe their success to the availability of large annotated datasets, and the Graphics Processing Units (GPUs) computational power and affordability. Furthermore, apart from developing successful deep models towards the aforementioned computer vision tasks, another research direction that flourishes during the recent few years while more efficient solutions are still striving to emerge, is the development of lightweight models capable of operating on devices with restricted computational resources such as mobile phones and embedded systems, [17, 18].

During the recent years, we have witnessed the successful introduction of Unmanned Aerial Vehicles (UAV), widely known as *drones*, in the media and entertainment industry. Drones have been applied in a wide spectrum of applications, ranging from entertainment to visual surveillance, rescue within the context of natural disasters [19], and medical emergencies [20], while previous practices in media production are gradually displaced due to their capability of capturing impressive aerial shots or shots of even inaccessible places. A major issue linked with the rise of drones is the demand for developing models for various computer vision tasks, able of both addressing the additional challenges of drone-captured images (such as small object size, unconstrained pose variations, occlusion), and running on-drone, that is with restricted processing power.

Thus, in this work, we propose regularized lightweight deep CNN models for various classification problems involved in autonomous systems, and more specifically, we consider binary classification problems in the context of media coverage of certain sport events (i.e. football match, bicycle race) by drones, allowing real-

time deployment for high resolution images. Furthermore, a key issue related to drone usage and autonomy is the demand for increased safety, since a drone may operate in vicinity of human crowds, and is potentially exposed to environmental hazards or unforeseeable errors that render their emergency landing inevitable. Hence, we also propose lightweight CNN models for crowd detection towards crowd avoidance. Finally, since face detection constitutes a primary step towards recognition of a specific bicyclist, or football player, we also deal with face detection. Summarizing, in this work we train CNN models for bicycle, crowd, face, football player detection in the context of media coverage of certain sport events by drones. Our goal is to produce semantic heatmaps [21] by e.g. predicting for each location within the captured high-resolution scene the crowd presence. That is, models with input of size either $32 \times 32 \times 3$ or $64 \times 64 \times 3$ which correspond to the width, height, and number of color channels of the input image, are trained. Then, test images are propagated to the network, and for every window 32×32 or 64×64 respectively the output of the network at the last convolutional layer is computed. An example of a crowd heatmap is provided in Fig. 1. Furthermore, the above procedure is useful in the camera control problem, [22]. That is, the semantic heatmaps for each of the aforementioned classification problems, aim to aid the algorithm for controlling the drone's camera for cinematography tasks by sending error signals. We should emphasize that the capability of handling high resolution images is extremely important for the application, since objects of interest in drone-captured images are of extremely small size, and thus image resizing, which is used by almost all of the state-of-the-art visual content analysis models (e.g. YOLO [9], SSD [8], etc.), would further shrink the object's size, rendering the detection impossible.

Subsequently, since we deal with lightweight models which usually perform worse than the more complex models, we aim at enhancing their performance. Thus, one objective of this paper, is to extensively study the impact of hinge loss against the cross entropy loss on the classification performance of binary classification problems. Additionally, we also perform experiments on multi-class datasets.

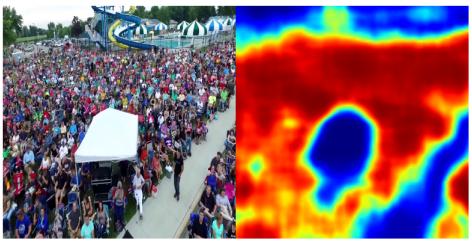


Figure 1: Crowded image and the corresponding predicted heatmap of crowd presence.

Finally, the third objective of this work is to propose a novel regularization method in order to reduce overfitting and improve the generalization ability of the utilized models. This is of considerable importance, in general, in deep learning, since neural networks are prone to overfitting due to their high capacity. In this work, we propose the so-called *Mutual Information (MI) regularizer*. The proposed regularizer is inspired by the Quadratic Mutual Information (QMI) measure [23], which is a variant of the commonly used Mutual Information, an information-theoretic measure of dependence between random variables. That is, apart from the classification loss, we propose to attach an additional optimization criterion based on the QMI. Recently, QMI reformulated to produce a kernel dimensionality reduction method under the Graph Embedding framework [24], while in [25] a Probabilistic Knowledge Transfer method proposed exploiting the QMI. We should note that the proposed regularization method is generic and can be applied in several deep learning architectures for classification purposes.

Over the past years, several regularization schemes have been proposed in order to improve weak generalization ability in neural networks, extended from common regularization methods, like the standard L1/L2 regularization which penalize large weights during the network optimization, to Dropout [26] where for each training sample, a randomly selected subset of the activations is set to zero in each epoch, and Dropconncet [27] that is a generalization of Dropout which instead of activations, sets a randomly selected subset of weights within the network to zero. Other earlier approaches include weight elimination, [28], and Bayesian methods, [29]. From another angle of view, multitask-learning [30] has

been employed as a way of enhancing the generalization ability of a model. For instance, techniques developed in semi-supervised learning were introduced in the deep learning domain, in [31] . That is, an unsupervised regularizer is combined with a supervised learner to perform semi-supervised learning. Furthermore, a novel CNN architecture with a SVM classifier at every hidden layer is proposed in [32]. This companion objective acts as a kind of feature regularization.

Summing up, the objective of this paper is to develop lightweight models for various classification tasks able to run in real time on-drone. We explore ways to enhance the classification performance of the lightweight models, first by investigating the effect of two widely used classification losses, that is the cross entropy and hinge losses, on the classification performance, and second by proposing a novel regularizer motivated by the QMI criterion.

The main contribution of this work can be summarized as follows:

- We propose regularized lightweight deep CNN models for various classification problems, capable of running in real-time on-drone.
- We empirically study the impact of hinge loss against cross entropy loss in binary classification problems and we argue that the hinge loss is better for binary classification problems.
- We propose a novel regularizer based on the QMI criterion in order to enhance the generalization ability of the utilized models.

The remainder of the manuscript is structured as follows. The utilized CNN architectures are described in Section 2. Section 3 presents the two compared loss functions. Subsequently, the proposed MI regularizer is presented in Section 4. The experiments, including the datasets description, the implementation details and the experimental results, are provided in Section 5. Finally, the conclusions are drawn in Section 6.

2. Lightweight CNN Models

In this Section, we provide the descriptions of the utilized architectures. A principal target of the utilized architectures is to permit real-time (that is about 25 frames per second) deployment on-drone for high resolution images. We should emphasize that it is critical for the application to handle high resolution images, since objects in drone-captured images are of extremely small size, and thus image resizing in order to meet real-time deployment limits, would further shrink the object of interest, rendering the detection impossible. Fig. 3 highlights the demand for high resolution images. That is, an aerial image that contains bicycles (bicycles with bicyclists) is provided, Fig. 3a, and the resulting heatmaps for input of various resolutions, i.e. 320×240, 480×360, 640×480, 1280×720, and 1920×1080, utilizing a proposed model, Figs. 3d-3h. As it shown, as the resolution increases, better performance can be accomplished. Furthermore, in Figs. 3b and 3c, the predictions for the same input, are provided, utilizing two state-of-the-art detectors which have been trained to detect among other classes, also persons and bicycles, that is YOLO v.2 [9] and Faster R-CNN [7], which operate for input 608×608 and 1000×600 respectively. As it is shown, both the state-of-the-art detectors perform poorly, while they also operate at much less than real-time, as it is shown below. It is also noteworthy, that SSD [8] and SSD with MobileNets [17], which operate for input 300×300, as well as for input 512×512, fail to detect any bicycle.

Thus, we utilize two architectures consisting of only five convolutional layers, by discarding the deepest layers and pruning filters of the widely used VGG-16 model [6]. That is, the first four convolutional layers of the VGG-16 model are used with pruned filters, while the last convolutional layer consists of two channels, each for a class, since we deal with binary classification problems. The first model can run in real-time on-drone for 720p (1280×720) resolution image and the second one can run in real-time for 1080p (1920×1080) resolution image. Thus, the models are abbreviated as VGG-720p and VGG-1080p, respectively, based on this attribute. Details on kernel sizes and channels of each layer of the

two architectures can be found in Table 1 and Table 2. The above descriptions of models concern input of training images of size 32×32. For input of size 64×64, using same kernels and channels, we use appropriate stride and pooling to achieve real-time deployment. Details on the utilized model architectures for both 32×32 and 64×64 input dimensions are depicted in Fig. 2.

The performance is evaluated on a low-power NVIDIA Jetson TX2 module with 8GB of memory, which is a state of the art GPU used for on-board drone perception. Furthermore, in order to accelerate the deployment speed and achieve real-time deployment, the TensorRT[1], deep learning inference optimizer is utilized. TensorRT is a library that optimizes deep learning models providing FP32 (default) and FP16 optimizations for production deployments of various applications. In Table 3, the detection speed in terms of frames per second (fps) is provided for the two architectures and their corresponding image resolution on the NVIDIA Jetson TX2 module without the utilization of the TensorRT optimizer, with the TensorRT on the default mode, as well as with TensorRT on the FP16 mode. As it shown, TensorRT and in particular the FP16 mode significantly accelerates the proposed models, achieving detection in-real time for high-resolution images.

Note that state-of-the-art detectors run at notably fewer fps on Jetson TX2, and also for lower resolution input images. For example, SSD [8] runs at 6 fps, for input of size 300×300, SSD with MobileNets [17] runs at 12.4 fps for the same input, and the Faster R-CNN [7] runs at 0.9 fps. Finally, YOLO v.2 [9] runs at 10 fps for input of size 308×308, while it runs at 3.1 fps for input of size 604×604 on the Jetson TX2 module. For fair comparisons, we also test the performance of YOLO v.2 utilizing TensorRT, as we have observed and it is also reported in literature [33, 34] that YOLO is faster than SSD and Faster-RCNN. Thus, utilizing the TensorRT optimizer YOLO v.2 runs at 7.8 fps for input of size 604×604, while further speed up is achieved with the FP16 mode up to 14.4 fps. It is noteworthy that state-of-the-art detectors like YOLO, can not achieve real-time detection on Jetson TX2, even utilizing the TensorRT optimizer, and also for

[1] https://developer.nvidia.com/tensorrt

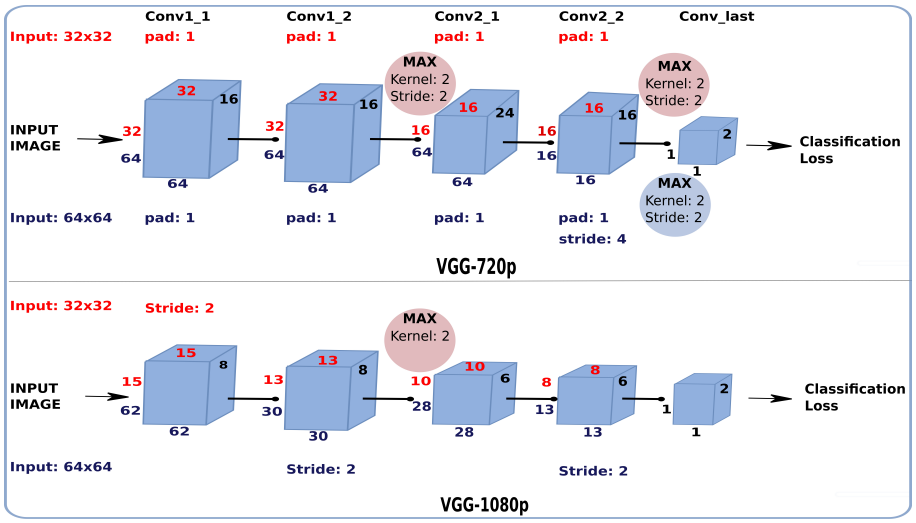


Figure 2: Model architectures: The VGG-720p model is depicted in the upper part of the figure, while the VGG-1080p model is depicted in the lower part of the figure. Details for input of size 32×32 are printed in red for both the model architectures, while details for input of size 64×64 are printed in blue.

considerably lower input resolution. Counterwise, the proposed models allow for real-time deployment utilizing TensorRT even for input resolution 1080p.

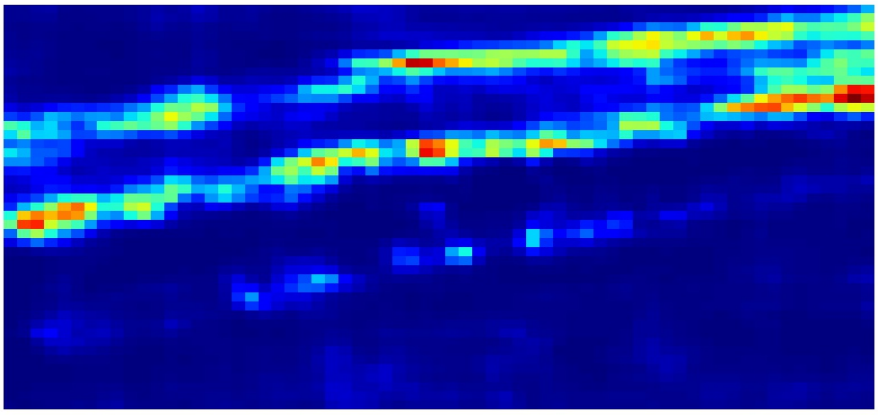
3. Hinge Loss Versus Cross Entropy Loss

Cross entropy loss and hinge loss functions are probably the most widely used loss functions in pattern classification. Support Vector Machines (SVM), [35], which use the hinge loss, constitute up to the present time a vivid research field [36, 37]. SVMs, which are inherently binary classifiers, seek for the optimal hyperplane which distinctly classifies the data samples, that is the hyperplane which maximizes the margin between the two classes. Considering multi-class classification problems several works have been proposed for formulating the SVM over multiple classes [38, 39, 40]. Amongst them, the earliest one and probably the most common technique is the one-against-all (or one-against-rest), which builds N_c one-against-all SVM models where N_c is the number of classes.

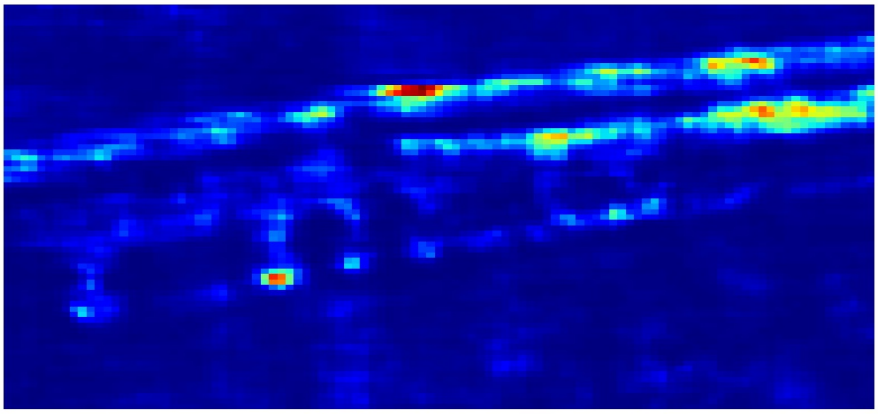
(a) Test image

(b) YOLO v.2 Prediction for input of size 604×604

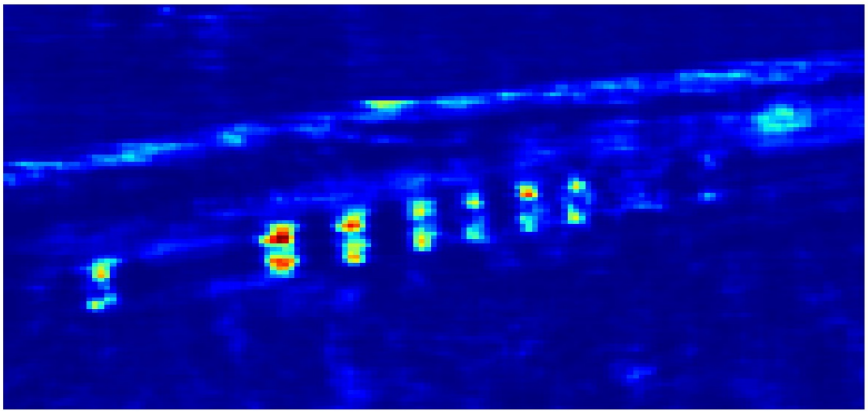
(c) Faster R-CNN Prediction for input of size 1000×600



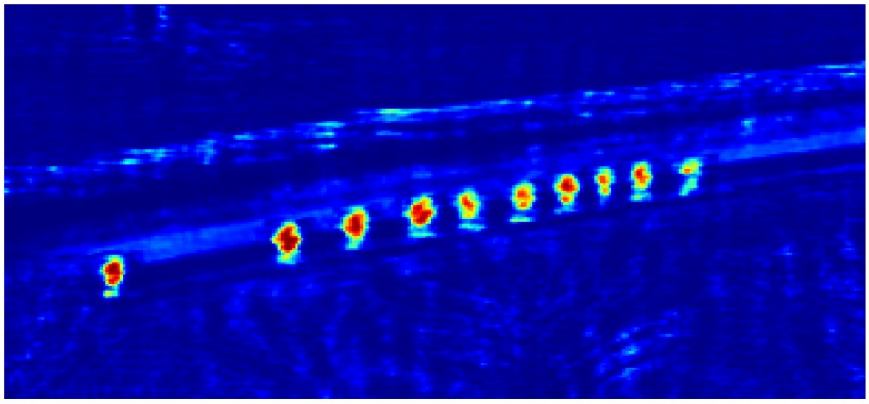
(d) Resulting heatmap for input of size 320×240, utilizing the proposed VGG-1080p model



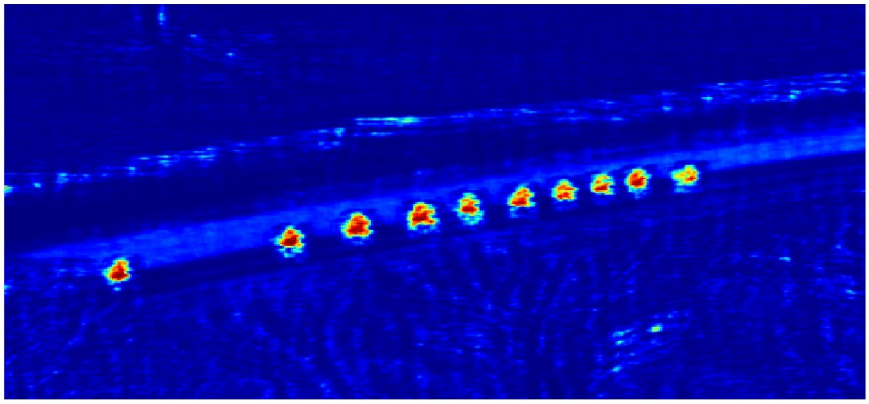
(e) Resulting heatmap for input of size 480×360, utilizing the proposed VGG-1080p model



(f) Resulting heatmap for input of size 640×480, utilizing the proposed VGG-1080p model



(g) Resulting heatmap for input of size 1280×720, utilizing the proposed VGG-1080p model



(h) Resulting heatmap for input of size 1920×1080, utilizing the proposed VGG-1080p model

Figure 3: An aerial high resolution image containing bicycles (3a), predictions utilizing the YOLO v2 and the Faster R-CNN detectors (3b)-(3c), and the resulting heatmaps for various deployment resolutions utilizing the proposed VGG-1080p model trained for bicycle detection (3d)-(3h).

Layer	Kernel	Channels
conv1_1	3×3	16
conv1_2	3×3	16
conv2_1	3×3	24
conv2_2	3×3	16
conv_last	8×8	2

Table 1: VGG-720p

Layer	Kernel	Channels
conv1_1	3×3	8
conv1_2	3×3	8
conv2_1	3×3	6
conv2_2	3×3	6
conv_last	8×8	2

Table 2: VGG-1080p

input	Model	Jetson TX2	TensorRT-FP32	TensorRT-FP16
32×32	VGG-720p	10.1	18.1	26.3
32×32	VGG-1080p	12.3	16.9	25.7
64×64	VGG-720p	8.7	16.6	25
64×64	VGG-1080p	8.8	18.5	25.6

Table 3: Speed (fps)

For a set of N input images $\mathcal{X} = \{\mathbf{X}_i, i = 1, \ldots, N\}$ we consider the corresponding scores with respect to each class, $\boldsymbol{y}_i^{last} \in \Re^{N_c \times 1}$. In the typical case the classification layer is implemented using a fully connected layer - with number of nodes equal to the number of classes - and the output is fed to the loss layer. In our case, since the objective is to develop lightweight models, and hence we propose fully convolutional architectures, instead of a fully connected layer, there is a convolutional layer with number of channels equal to the number of classes and kernel with receptive field equal to the whole input volume.

Then, the hinge loss is defined as:

$$L_{hinge} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N_c} max\Big(0, 1 - \delta\{c_i = j\}\ y_{i,j}^{last}\Big), \tag{1}$$

where $c_i \in [1, \ldots, N_c]$ indicates the correct class among the N_c classes, $y_{i,j}^{last}$ indi-

cates the score with respect to the j-th class for the i-th image, and

$$\delta\{condition\} = \begin{cases} 1 & \text{, if condition} \\ -1 & \text{, otherwise} \end{cases}$$

In this work, we deal with binary classification problems. That is, $N_c = 2$.

Cross entropy loss or softmax classifier, is extensively used in deep learning architectures [5, 6, 41], providing an intuitive output of normalized class probabilities. Instead of computing scores for each class, like the SVM classifer, the softmax classifier computes the scores for each class, and then applies the softmax function [42] to transform them to a vector of values between zero and one that sum to one, in order to be interpreted as class probabilities. Finally, the classification process is realized using the cross entropy loss function.

The cross entropy loss is defined as:

$$L_{cross\_entropy} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N_c} l_{i,j} log(p_{i,j}), \tag{2}$$

where N_c is the number of classes, $l_{i,j} \in \{0, 1\}$ is a binary indicator that takes the value 1 if the the sample i belongs to class j , and $p_{i,j}$ is the predicted softmax probability the sample i to belong to class j. For a two-class classification problem the cross entropy loss can be calculated as:

$$L_{cross\_entropy\_binary} = -\frac{1}{N}\sum_{i=1}^{N}\Big(l_i log(p_i) + (1 - l_i) log(1 - p_i)\Big) \tag{3}$$

To the best of our knowledge, there is no other previous work extensively investigating the impact of hinge loss against cross entropy loss on the classification accuracy, with special emphasis to binary classification problems. Surveying the relevant literature, we observe that the cross entropy loss is widely used in deep CNNs for dealing with multi-class classification problems [5, 6, 41].

On the other hand, in [43] the author first proposes to replace the softmax loss layer (i.e. cross entropy loss), with a linear SVM layer. Particularly, the L2-SVM objective [44] is utilized instead of the standard hinge loss. Experimental results on MNIST-10 and CIFAR-10 datasets show that for some deep neural models, the linear SVM layer is beneficial over the softmax loss one.

In [45] the authors provide comparisons among various classification losses, including the cross entropy and hinge losses, for multi-class classification problems. The authors conclude that depending on the application of the deep model, losses other than cross entropy loss are preferable. Subsequently, in [32], which is a work with a different goal where a new CNN architecture with a SVM classifier at each hidden layer is proposed, we also observe that a CNN with SVM loss layer outperforms the CNN with softmax loss layer in the MNIST-10 dataset.

Finally, studying the work presented in [46], and particularly in the Feature Analysis section where an analysis of the discriminative information of each layer is provided, apart from the stated observations on the importance of the model's depth, some potentially interesting remarks arise. That is, we first observe that in a dataset with comparatively few classes the SVM classifier outperforms the softmax classifier, while in a similar dataset with much more classes, the softmax classifier performs better. Furthermore, we see that in the first dataset of fewer classes, the difference between the SVM classifier over the softmax is notably bigger in the less deep layer. Thus, this also enhances our motivation to investigate the efficiency of hinge loss as compared to the cross entropy loss, since a major objective of this work is to provide lightweight models with improved performance.

In order to obtain real-time performance someone has to severely decrease the number of layers and the number of filters. Thus, the resulting lightweight models are weaker than the heavier ones in terms of performance. In order to improve their performance we should exploit the available losses and possible regularizers that fit better to the specific tasks, which is binary classification with limited computational resources.

4. The Proposed MI Regularizer

In this paper, we propose a novel regularizer motivated by the Quadratic Mutual Information [23]. Apart from the classification loss, we propose a regularization loss derived from the so-called information potentials of the QMI. Thus, in

this Section, we first introduce the Mutual Information and its quadratic variant, and then we present the proposed MI regularizer.

We assume a random variable Y representing the image representations of the feature space generated by a specific deep neural layer. We also assume a discrete-value variable C that represents the class labels. For each feature represenation $\boldsymbol{y}$ there is a class label c. The MI measures dependence between random variables, first introduced by Shannon, [47]. That is, the MI measures how much the uncertainty for the class label c is reduced by observing the feature vector $\boldsymbol{y}$. Let $p(c)$ be the probability of observing the class label c, and $p(\boldsymbol{y}, c)$ the probability density function of the corresponding joint distribution.

The MI between the two random variables is defined as:

$$MI(Y, C) = \sum_{c} \int_{\boldsymbol{y}} p(\boldsymbol{y}, c) \log \frac{p(\boldsymbol{y}, c)}{p(\boldsymbol{y})P(c)} d\boldsymbol{y}, \tag{4}$$

where $P(c) = \int_{\boldsymbol{y}} p(\boldsymbol{y}, c) d\boldsymbol{y}$. MI can also be interpreted as a Kullback-Leibler divergence between the joint probability density $p(\boldsymbol{y}, c)$ and the product of marginal probabilities $p(\boldsymbol{y})$ and $P(c)$.

QMI is derived by replacing the Kullback-Leibler divergence by the quadratic divergence measure [23]. That is:

$$QMI(Y, C) = \sum_{c} \int_{\boldsymbol{y}} \left(p(\boldsymbol{y}, c) - p(\boldsymbol{y})P(c)\right)^2 d\boldsymbol{y}. \tag{5}$$

And thus, by expanding eq. (5) we arrive at the following equation:

$$QMI(Y, C) = \sum_{c} \int_{\boldsymbol{y}} p(\boldsymbol{y}, c)^2 d\boldsymbol{y} + \sum_{c} \int_{\boldsymbol{y}} p(\boldsymbol{y})^2 P(c)^2 d\boldsymbol{y} - 2 \sum_{c} \int_{\boldsymbol{y}} p(\boldsymbol{y}, c) p(\boldsymbol{y}) P(c) d\boldsymbol{y}. \tag{6}$$

The quantities appearing in eq. (6), are called *information potentials* and they are defined as follows: $V_{IN} = \sum_c \int_{\boldsymbol{y}} p(\boldsymbol{y}, c)^2 d\boldsymbol{y}$, $V_{ALL} = \sum_c \int_{\boldsymbol{y}} p(\boldsymbol{y})^2 P(c)^2 d\boldsymbol{y}$, $V_{BTW} = \sum_c \int_{\boldsymbol{y}} p(\boldsymbol{y}, c) p(\boldsymbol{y}) P(c) d\boldsymbol{y}$, and thus, the quadratic mutual information between the data samples and the corresponding class labels can be expressed as follows in terms of the information potentials:

$$QMI = V_{IN} + V_{ALL} - 2V_{BTW}. \tag{7}$$

If we assume that there are N_c different classes, each of them consisting of J_p samples, the class prior probability for the c_p class is given as: $P(c_p) = \frac{J_p}{N}$, where N corresponds to the total number of samples. Kernel Density Estimation [48] can be used to estimate the joint density probability: $p(\boldsymbol{y}, c_p) = \frac{1}{N}\sum_{j=1}^{J_p} K(\boldsymbol{y}, \boldsymbol{y}_{pj}; \sigma^2)$, for a symmetric kernel K, with width σ, where we use the notation $\boldsymbol{y}_{pj}$ to refer to the j-th sample of the p-th class, as well as the probability density of Y as $p(\boldsymbol{y}) = \sum_{p=1}^{J_p} p(\boldsymbol{y}, c_p) = \frac{1}{N}\sum_{j=1}^{N} K(\boldsymbol{y}, \boldsymbol{y}_j; \sigma^2)$.

Thus, eq. (7) is formulated as follows:

$$V_{IN} = \frac{1}{N^2}\sum_{p=1}^{N_c}\sum_{k=1}^{J_p}\sum_{l=1}^{J_p} K(\boldsymbol{y}_{pk}, \boldsymbol{y}_{pl}; 2\sigma^2), \tag{8}$$

$$V_{ALL} = \frac{1}{N^2}\Big(\sum_{p=1}^{N_c}(\frac{J_p}{N})^2\Big)\sum_{k=1}^{N}\sum_{l=1}^{N} K(\boldsymbol{y}_k, \boldsymbol{y}_l; 2\sigma^2), \tag{9}$$

$$V_{BTW} = \frac{1}{N^2}\sum_{p=1}^{N_c}\frac{J_p}{N}\sum_{j=1}^{J_p}\sum_{k=1}^{N} K(\boldsymbol{y}_{pj}, \boldsymbol{y}_k; 2\sigma^2). \tag{10}$$

The kernel function $K(\boldsymbol{y}_i, \boldsymbol{y}_j; \sigma^2)$ expresses the similarity between two samples i and j. There are several choices for the kernel function, [48]. For example, in [23] the Gaussian kernel is used, while in [25] the authors utilize a cosine similarity based kernel to avoid defining the width, in order to ensure that a meaningful probability estimation is obtained, since finetuning the width of the kernel is not a straightforward task, [49]. In our experiments, we use as kernel metric a Euclidean based similarity, which also absolves us from defining the width of the kernel. Given two vectors $\boldsymbol{y}_i$, $\boldsymbol{y}_j$, the Gaussian kernel is defined as:

$$K_G = \exp\Big(-\frac{||\boldsymbol{y}_i - \boldsymbol{y}_j||_2^2}{\sigma}\Big). \tag{11}$$

And the Euclidean-based similarity is defined as:

$$K_{ED} = \frac{1}{1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||_2^2}. \tag{12}$$

The pairwise interactions described above between the samples can be interpreted as follows:

- V_{IN} expresses the interactions between pairs of samples inside each class
- V_{ALL} expresses the interactions between all pairs of samples, regardless of the class membership
- V_{BTW} expresses the interactions between samples of each class against all other samples

Thus, motivated by the QMI, in this work we propose a novel regularizer in order to enhance the generalization ability of a deep model. That is, apart from the optimization criterion defined by the hinge loss function which aims at separating the samples belonging to different classes, we propose an additional optimization criterion utilizing the information potential defined in eq. (7). We assume that the hinge loss preserves the V_{BTW} information potential which aims to separate samples belonging to different classes. Then, our objective is to maximize pairwise interactions between the samples described by the $V_{IN} + V_{ALL}$ quantities. The derived joint optimization criterion defines an additional loss function, which is attached to the penultimate convolutional layer (that is the last convolutional layer, before the one utilized for the classification task) and acts as regularizer to the main classification objective.

$$L_{MI} = -(V_{IN} + V_{ALL}), \tag{13}$$

where:

$$V_{IN} = \frac{1}{N^2} \sum_{p=1}^{N_c} \sum_{k=1}^{J_p} \sum_{l=1}^{J_p} K_{ED}(\boldsymbol{y}_{pk}, \boldsymbol{y}_{pl}), \tag{14}$$

and

$$V_{ALL} = \frac{1}{N^2} \Big(\sum_{p=1}^{N_c} (\frac{J_p}{N})^2 \Big) \sum_{k=1}^{N} \sum_{l=1}^{N} K_{ED}(\boldsymbol{y}_k, \boldsymbol{y}_l). \tag{15}$$

Considering binary classification problems the above optimization criteria can be formulated as follows:

$$V_{IN} = \frac{1}{N^2} \sum_{k=1}^{J_1} \sum_{l=1}^{J_1} K_{ED}(\boldsymbol{y}_{1k}, \boldsymbol{y}_{1l}) + \frac{1}{N^2} \sum_{k=1}^{J_2} \sum_{l=1}^{J_2} K_{ED}(\boldsymbol{y}_{2k}, \boldsymbol{y}_{2l}), \tag{16}$$

and

$$V_{ALL} = \frac{1}{N^2}\left(\frac{J_1^2 + J_2^2}{N^2}\right)\sum_{k=1}^{N}\sum_{l=1}^{N} K_{ED}(\boldsymbol{y}_k, \boldsymbol{y}_l), \tag{17}$$

The total loss for the network training is defined as:

$$L_{total} = L_{Hinge} + \eta L_{MI}, \tag{18}$$

where the parameter $\eta \in [0, 1]$ controls the relative importance of L_{MI}. We solve the above optimization problem using gradient descent. We should note that the proposed regularizer can be applied for the whole dataset, as well as in terms of mini-batch training. In our experiments we utilize the mini-batch mode. We should finally note that in the regularized training we utilize the hinge loss layer since, as we show, it performs steadily better than the cross entropy one in binary classification problems, however the cross entropy loss could also be utilized.

5. Experiments

In this Section, we present the experiments conducted in order to evaluate the impact of hinge loss against cross entropy loss in the classification performance, as well as the effectiveness of the proposed regularizer in improving the classification performance. To this aim, we first performed experiments on six datasets, four two-class datasets constructed for various classification problems involved in the context of media coverage of specific sport events by drones, and two multiple-class datasets. Subsequently, in order to evaluate the performance of the MI regularizer, we performed experiments on the aforementioned two-class datasets. The descriptions of the utilized datasets are presented in the following subsections. Two post-hoc Bonferroni tests conducted in order to evaluate the statistical significance of the obtained results. Finally, qualitative results are provided utilizing real world drone images for evaluating the performance of the proposed models trained with the MI regularizer. Throughout this work, we use Test Accuracy (Classification Accuracy) to evaluate the proposed method. Each experiment is repeated five times and we report the mean value and the standard deviation, considering the maximum value of Test Accuracy for each experiment.

The probabilistic factor is the random weight initialization. We also provide the curves of mean Test Accuracy.

5.1. Face

The dataset contains 70,000 train images of faces and equal number of train images of non-faces, and a test set of 7,468 images. Images of faces have been randomly selected from the AFLW [50], MTFL [51], and WIDER FACE [52] datasets. Input images are of size 32×32. Sample images of the constructed *Face* dataset are presented in Fig. 4.

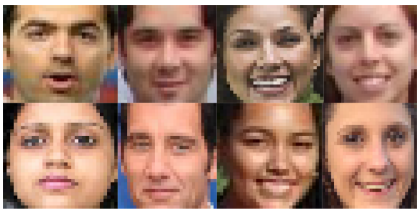


Figure 4: Sample images of the Face dataset.

5.2. Football Player

The Football Player dataset, [21] consists of 98,000 train images that contain football players and non-football players, and a test set of 10,000 images. Input images are of size 32×32. Sample images of the *Football Player* dataset are illustrated in Fig. 5.

Figure 5: Sample images of the Football Player dataset.

5.3. Crowd-Drone

The dataset contains 40,000 train images of crowded scenes and non-crowded scenes, and 11,550 test images. Input images are of size 64×64. Sample images

of the constructed *Crowd-Drone* dataset are presented in Fig. 6.

Figure 6: Sample images of the Crowd-Drone dataset.

5.4. Bicycles

The Bicycles dataset, [21] contains 51,200 equally distributed train images of bicycles (bicycle with bicyclist) and non-bicycles, and a test set of 10,000 images. Input images are of size 64×64. Sample images of the constructed *Bicycle* dataset are presented in Fig. 7.

Figure 7: Sample images of the Bicycles dataset.

5.5. Street View House Numbers

The Street View House Numbers (SVHN) dataset, [53], obtained from house numbers in Google Street View images. It contains 73,257 train images and 26,032 test images, divided into 10 classes, 1 for each digit from 0 to 9. Input images are of size 32×32 and sample images are provided in Fig. 8

Figure 8: Sample images of the SVHN dataset.

5.6. *Cifar-10*

The *Cifar-10* dataset, [54], consists of 60,000 images of size 32×32 divided into 10 classes with 6,000 images per class. 50,000 images are used as the train set and 10,000 images as the test set. Sample images of the *Cifar-10* dataset are provided in Fig. 9

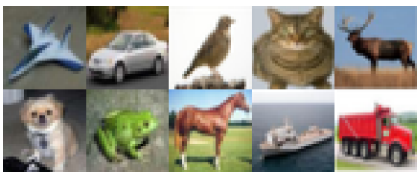


Figure 9: Sample images of the Cifar-10 dataset.

5.7. *Implementation Details*

All the experiments conducted using the Caffe Deep Learning framework [55]. We use the mini-batch gradient descent for the networks' training. That is, an update is performed for every mini-batch of N_b training samples. The learning rate is set to 10^{-3} and drops to 10^{-4} gradually, and the batch size is set to 256. The momentum is 0.9. All the models are trained on an NVIDIA GeForce GTX 1080 with 8GB of GPU memory for 100 epochs, and can run in real-time when deployed on an NVIDIA Jetson TX2. Regarding the parameter which controls the importance of the regularization term of common regularizers, like L1 and L2, is usually set to 0.0005. In this work, the parameter η in (18) which controls the relative importance of the proposed regularizer's loss, is set to 0.001, since we have seen that in most cases provides best performance. However we should

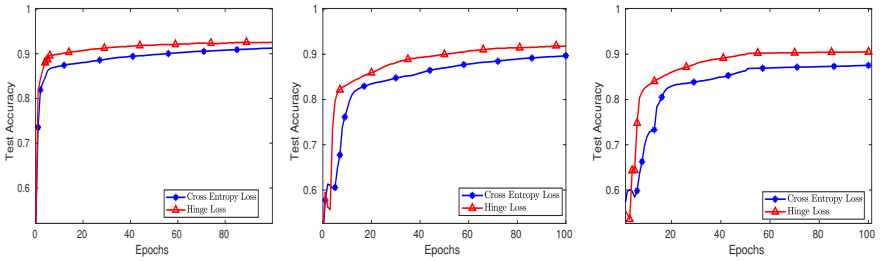
note the proposed reularizer improves the performance for different values of η, too. For example, in Table 4, we representatively provide the experimental results for various values of η, in the case of the Bicycles dataset, utilizing the VGG-720p model. As we can see, the MI regularizer operates improvingly in any case. The comparisons among hinge loss against cross entropy loss, as well as only hinge loss against hinge loss & MI regularizer performed with the exact same training settings. Note that we have set the learning rate to 10^{-3} (with drop policy) since it appears to be generally the most appropriate. However, experiments also performed with various values of learning rate (LR). In Table 5, we provide some indicative results in the Face dataset in terms of Test Accuracy, where the superiority of hinge loss over cross entropy loss is verified for various learning rates, while in Fig. 10 we present the corresponding mean Test Accuracy over the 100 epochs of training. Best results are printed in bold.

Training Approach	η	**Test Accuracy**
Only Hinge Loss	-	0.9785 ± 0.0021
Hinge Loss & MI Regularizer	1	0.9814 ± 0.0015
Hinge Loss & MI Regularizer	0.1	0.9827 ± 0.0018
Hinge Loss & MI Regularizer	0.01	0.9836 ± 0.002
Hinge Loss & MI Regularizer	0.001	**0.9884 ± 0.0011**

Table 4: Bicycle Dataset - VGG-720p: Impact of parameter η in eq. (18)

LR	**Cross Entropy Loss**	**Hinge Loss**
$LR = 10^{-3}$ - drop	0.9126 ± 0.0021	**0.9273 ± 0.0036**
$LR = 10^{-4}$ - fixed	0.896 ± 0.0025	**0.9197 ± 0.0046**
$LR = 10^{-4}$ - drop	0.8755 ± 0.004	**0.9054 ± 0.0014**

Table 5: Face Dataset - VGG-720p - Test Accuracy for various learning rates



(a) Face: VGG-720p - $LR = 10^{-3}$ - drop (b) Face: VGG-720p - $LR = 10^{-4}$ - fixed (c) Face: VGG-720p - $LR = 10^{-4}$ - drop

Figure 10: Face Dataset - VGG-720p - Various learning rates

5.8. Experimental Results

In the first set of experiments, we evaluate the classification performance of hinge loss against cross entropy loss. In Fig. 11 we provide the comparison of the mean Test Accuracy of the hinge loss against cross entropy loss, utilizing both the proposed models for all the two-class datasets, that is the Face, Football Player, Crowd-Drone, and Bicycles datasets. Furthermore, in Tables 8-11 we present the mean value and the standard deviation of the Test Accuracy, for the two losses under consideration. As we can observe the hinge loss is steadily superior over the cross entropy loss.

In Tables 6 and 7 we provide the corresponding evaluation results on the SVHN and Cifar-10 datasets, for the proposed VGG-720p model. We should note that the VGG-1080p architecture could not achieve remarkable classification performance on the aforementioned multi-class datasets. As we can observe, the cross entropy loss achieves marginally superior performance in the SVHN dataset, whilst the hinge loss outperforms the cross entropy one in the Cifar-10 dataset. Thus, we could observe that the hinge loss is undoubtedly the optimal choice for binary classification problems, considering lightweight deep models, however this is not a safe claim in multi-class classification problems.

In the second set of experiments, we evaluate the proposed MI regularizer. In Tables 8-11 we present the mean value and the standard deviation of the Test Accuracy, for the considered training approaches, that is utilizing only cross entropy

loss, only hinge loss, and hinge loss with the proposed MI regularizer, utilizing both the proposed models. Correspondingly, in Figs. 12-15 we illustrate the curves of mean Test Accuracy of the only hinge loss training against hinge loss & MI regularized training. We can see in the demonstrated results, that the proposed MI regularizer remarkably enhances the classification performance for both the proposed model architectures on all the utilized datasets. In Table 12 we provide representative comparisons against the common L1 and L2 regularizers on the Bicycles dataset, utlizing the VGG-720p model. As we can see, the L2 regularizer marginally improves the performance and the L1 one harms the perfomance, while the proposed MI regularizer achieves considerably better performance. Finally, it is noteworthy that we have tested the performance of the proposed MI regularizer on additional non-real-time models, where its effectiveness is further validated. However, we do not include these experiments, since a principal target of this work is to provide real-time models.

Training Approach	**VGG-720p**
Cross Entropy Loss	**0.8987 ± 0.0019**
Hinge Loss	0.8972 ± 0.0057

Table 6: SVHN-10 Dataset - Test Accuracy

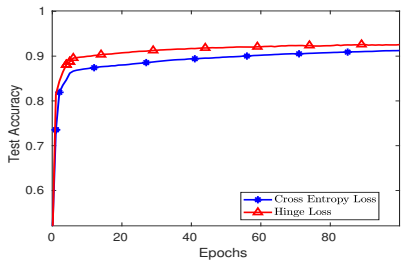
Training Approach	**VGG-720p**
Cross Entropy Loss	0.5801 ± 0.0161
Hinge Loss	**0.5919 ± 0.016**

Table 7: CIFAR-10 Dataset - Test Accuracy

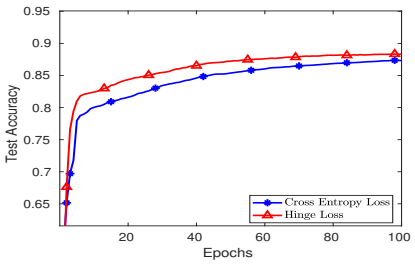
In the third set of experiments, we conducted two post-hoc Bonferroni tests [56], first for ranking the hinge loss and cross entropy loss for binary classification problems and evaluating the statistical significance of the obtained results, and second for ranking the proposed regularization method and the only hinge loss training and evaluating the statistical significance of the obtained results. The

Training Approach	**VGG-720p**	**VGG-1080p**
Only Cross Entropy Loss	0.9126 ± 0.0021	0.8738 ± 0.0052
Only Hinge Loss	0.9273 ± 0.0036	0.8841 ± 0.004
Hinge Loss & MI Regularizer	**0.9292 ± 0.0048**	**0.8896 ± 0.0007**

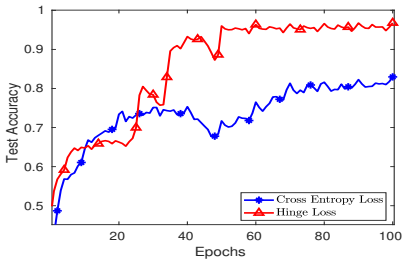
Table 8: Face Dataset - Test Accuracy



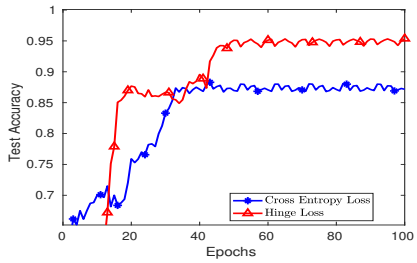
(a) Face: VGG-720p



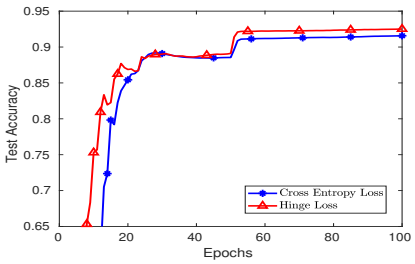
(b) Face: VGG-1080p



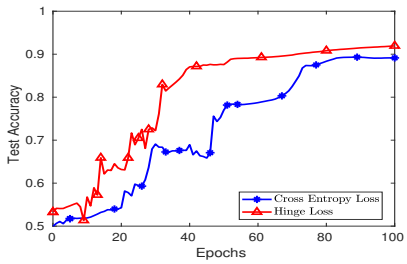
(c) Football Player: VGG-720p



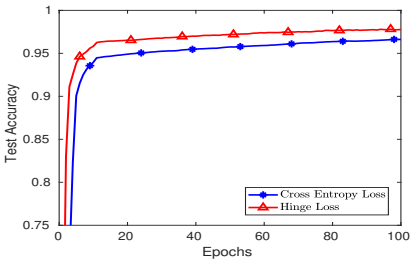
(d) Football Player: VGG-1080p



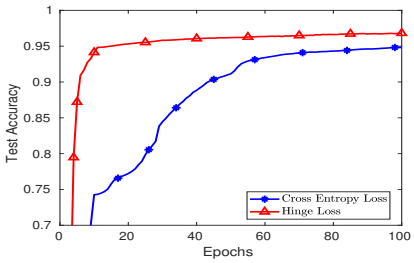
(e) Crowd-Drone: VGG-720p



(f) Crowd-Drone: VGG-1080p



(g) Bicycles: VGG-720p



(h) Bicycles: VGG-1080p

Figure 11: Cross Entropy Loss VS Hinge Loss

Training Approach	VGG-720p	VGG-1080p
Only Cross Entropy Loss	0.9183 ± 0.0307	0.8897 ± 0.0747
Only Hinge Loss	0.9680 ± 0.0080	0.9568 ± 0.01
Hinge Loss & MI Regularizer	**0.9813 ± 0.0027**	**0.9744 ± 0.01**

Table 9: Football Player Dataset - Test Accuracy

Training Approach	VGG-720p	VGG-1080p
Only Cross Entropy Loss	0.9157 ± 0.0058	0.9030 ± 0.014
Only Hinge Loss	0.9334 ± 0.01	0.9194 ± 0.0082
Hinge Loss & MI Regularizer	**0.9371 ± 0.0011**	**0.9303 ± 0.0076**

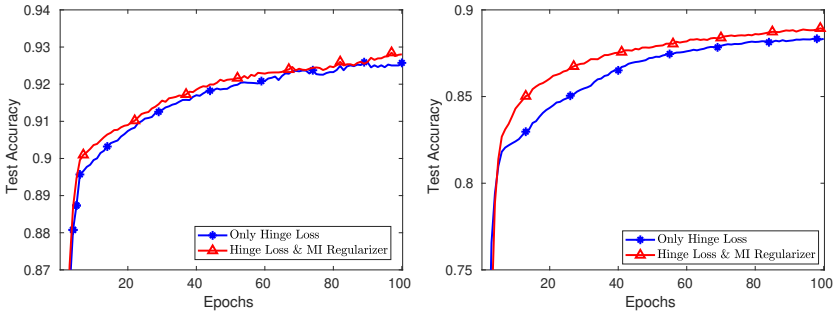
Table 10: Crowd-Drone Dataset - Test Accuracy

Training Approach	VGG-720p	VGG-1080p
Only Cross Entropy Loss	0.9664 ± 0.001	0.9484 ± 0.0023
Only Hinge Loss	0.9785 ± 0.0021	0.9684 ± 0.0037
Hinge Loss & MI Regularizer	**0.9884 ± 0.0011**	**0.9696 ± 0.0018**

Table 11: Bicycle Dataset - Test Accuracy

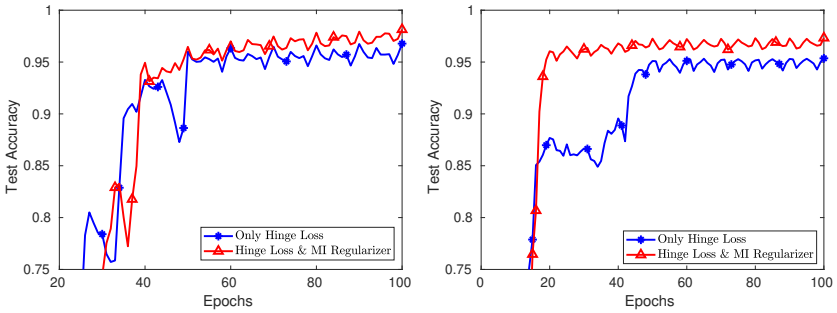
Training Approach	Test Accuracy
Only Hinge Loss	0.9785 ± 0.0021
Hinge Loss & MI Regularizer	**0.9884 ± 0.0011**
Hinge Loss & L1 Regularizer	0.9719 ± 0.0027
Hinge Loss & L2 Regularizer	0.9797 ± 0.001

Table 12: Bicycles Dataset - VGG-720p: Comparison against the common L1 and L2 regularizers



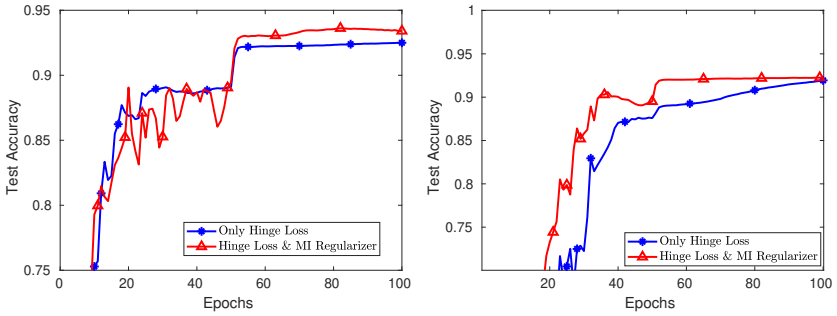
(a) VGG-720p (b) VGG-1080p

Figure 12: Face Dataset: MI Regularizer



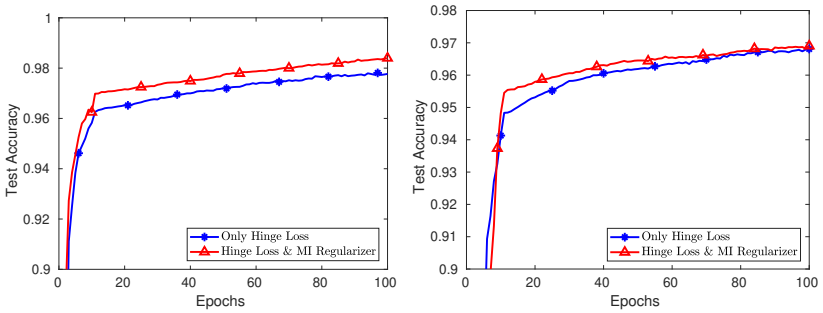
(a) VGG-720p (b) VGG-1080p

Figure 13: Football Player Dataset: MI Regularizer



(a) VGG-720p (b) VGG-1080p

Figure 14: Crowd-Drone Dataset: MI Regularizer



(a) VGG-720p (b) VGG-1080p

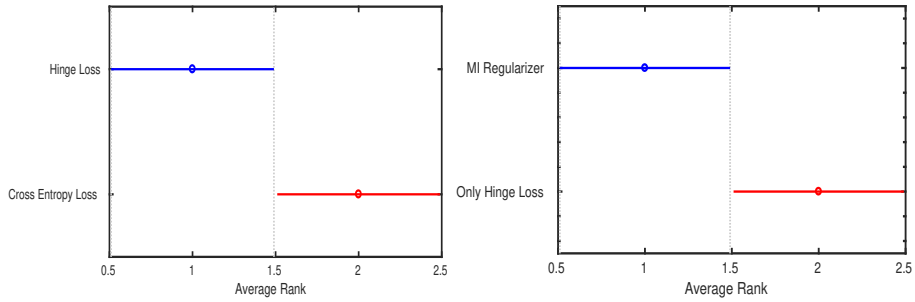
Figure 15: Bicycles Dataset: MI Regularizer

performance of two methods is significantly different, if the corresponding average ranks over the datasets differ by at least the critical difference:

$$CD = q_a \sqrt{\frac{m(m+1)}{6D}}, \tag{19}$$

where m is the number of methods compared, D is the number of datasets and critical values q_α can be found in [56]. In our comparisons we set $\alpha = 0.05$. The number of datasets is four in the performed tests. The compared methods are two, that is the training using the hinge loss is compared with a control method which is the training with cross entropy loss, and second the proposed regularizer is compared with a control method which is the only hinge loss training approach. The ranking results are illustrated in Figs. 16a and 16b, respectively. The vertical axis depicts the two methods, while the horizontal axis depicts the performance ranking. The circles indicate the mean rank and the intervals around them indicate the confidence interval as this is determined by the CD value. Overlapping intervals between two methods indicate that there is not a statistically significant difference between the corresponding ranks, while non-overlapping intervals indicate that the compared methods are significantly different. As we can observe, the hinge loss is significantly different against cross entropy loss for binary classification problems, as well as the proposed regularizer is significantly different against the only hinge loss training approach. We should note that we representatively present the performance utilizing the VGG-720p architecture, however identical

performance is achieved utilizing the VGG-1080p architecture.



(a) Cross Entropy Loss vs Hinge Loss for Binary Classification

(b) MI regularizer vs Only Hinge Loss

Figure 16: Post-Hoc Bonferroni Tests

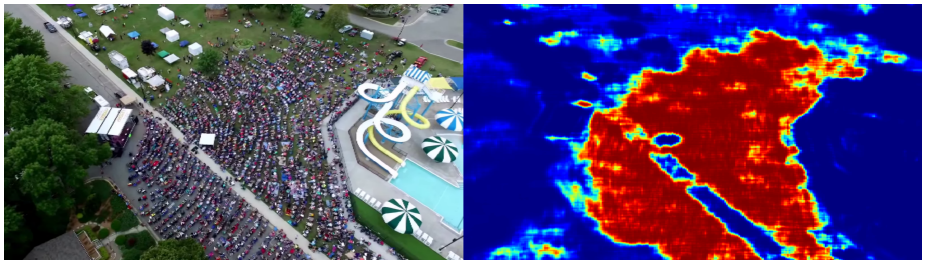
Finally, in the forth set of experiments, we compute heatmaps of the object's presence for the classification problems under consideration, so as to provide some qualitative results for evaluating the effectiveness of the proposed real-time models trained with the MI regularizer. Thus, considering high resolution images captured by drones, the heatmaps considering the tasks of bicycle, football player, and crowd detection utilizing the proposed VGG-1080p models are computed. Evaluation results provided in Fig. 17 indicate the efficiency of the proposed models.

6. Conclusions

In this paper, we proposed regularized lightweight CNN models for addressing various classification tasks (e.g. crowd detection, bicycle detection, etc.) able to run in real-time on-drone. Second, in order to explore ways to enhance the classification performance of the lightweight models, we extensively investigated the impact of two widely used classification losses, that is the cross entropy and hinge losses, on the classification performance, considering binary classification problems. Third, we proposed a novel regularizer motivated by the QMI, the so-called MI regularizer. The performance was evaluated on four datasets.

(a) Bicycle Detection

(b) Football Player Detection



(c) Crowd Detection

Figure 17: Heatmaps on real world drone images for specific detection problems utilizing the corresponding proposed models.

The evaluation results indicate that hinge loss is undoubtedly the optimal choice for binary classification problems, considering lightweight deep models. Furthermore, the effectiveness of the proposed regularizer in enhancing the generalization ability of the proposed models is also validated. Finally, even the proposed models can achieve significant performance in the considered two-class classification problems, it can be observed in the experimental results that there is a drop performance in more complex problems (i.e Cifar-10 dataset). Thus, future work will

consist on developing more efficient models, capable of efficiently addressing, in terms of both speed and accuracy, more complicated problems under the considered computation and memory constraints. A first step towards this goal is to investigate if we can incorporate established methodologies (e.g. skip connections [41]) to our work.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

[1] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

[2] J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks 61 (2015) 85–117.

[3] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in neural information processing systems 2, Morgan Kaufmann Publishers Inc., 1990, pp. 396–404.

[4] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognition 77 (2018) 354–377.

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[6] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[7] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[9] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, arXiv preprint arXiv:1612.08242.

[10] S. Pang, J. Zhu, J. Wang, V. Ordonez, J. Xue, Building discriminative cnn image representations for object retrieval using the replicator equation, Pattern Recognition.

[11] M. Tzelepi, A. Tefas, Deep convolutional learning for content based image retrieval, Neurocomputing 275 (2018) 2467–2478.

[12] M. Tzelepi, A. Tefas, Deep convolutional image retrieval: A general framework, Signal Processing: Image Communication 63 (2018) 30–43.

[13] P. Li, D. Wang, L. Wang, H. Lu, Deep visual tracking: Review and experimental comparison, Pattern Recognition 76 (2018) 323–338.

[14] B. Chen, P. Li, C. Sun, D. Wang, G. Yang, H. Lu, Multi attention module for visual tracking, Pattern Recognition.

[15] E. Daskalakis, M. Tzelepi, A. Tefas, Learning deep spatiotemporal features for video captioning, Pattern Recognition Letters 116 (2018) 143–149.

[16] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size, arXiv preprint arXiv:1602.07360.

[19] L. Apvrille, T. Tanzi, J.-L. Dugelay, Autonomous drones for assisting rescue services within the context of natural disasters, in: General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI, IEEE, 2014, pp. 1–4.

[20] A. Claesson, D. Fredman, L. Svensson, M. Ringh, J. Hollenberg, P. Nordberg, M. Rosenqvist, T. Djarv, S. Osterberg, J. Lennartsson, et al., Unmanned aerial vehicles (drones) in out-of-hospital-cardiac-arrest, Scandinavian journal of trauma, resuscitation and emergency medicine 24 (1) (2016) 124.

[21] M. Tzelepi, A. Tefas, Graph embedded convolutional neural networks in human crowd detection for drone flight safety, IEEE Transactions on Emerging Topics in Computational Intelligence.

[22] N. Passalis, A. Tefas, I. Pitas, Efficient camera control using 2d visual information for unmanned aerial vehicle-based cinematography, in: Circuits and Systems (ISCAS), 2018 IEEE International Symposium on, IEEE, 2018, pp. 1–5.

[23] K. Torkkola, Feature extraction by non-parametric mutual information maximization, Journal of machine learning research 3 (Mar) (2003) 1415–1438.

[24] D. Bouzas, N. Arvanitopoulos, A. Tefas, Graph embedded nonparametric mutual information for supervised dimensionality reduction, IEEE transactions on neural networks and learning systems 26 (5) (2015) 951–963.

[25] N. Passalis, A. Tefas, Learning deep representations with probabilistic knowledge transfer, in: The European Conference on Computer Vision (ECCV), 2018.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[27] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using dropconnect, in: International Conference on Machine Learning, 2013, pp. 1058–1066.

[28] A. S. Weigend, D. E. Rumelhart, B. A. Huberman, Generalization by weight-elimination with application to forecasting, in: Advances in neural information processing systems, 1991, pp. 875–882.

[29] D. J. MacKay, Probable networks and plausible predictionsâ"a review of practical bayesian methods for supervised neural networks, Network: Computation in Neural Systems 6 (3) (1995) 469–505.

[30] R. Caruana, Multitask learning, Machine learning 28 (1) (1997) 41–75.

[31] J. Weston, F. Ratle, R. Collobert, Deep learning via semi-supervised embedding, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 1168–1175.

[32] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Artificial Intelligence and Statistics, 2015, pp. 562–570.

[33] S. Zhang, L. Wen, X. Bian, Z. Lei, S. Z. Li, Single-shot refinement neural network for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4203–4212.

[34] P. Nousi, E. Patsiouras, A. Tefas, I. Pitas, Convolutional neural networks for visual information analysis with limited computing resources, in: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 321–325.

[35] V. N. Vapnik, V. Vapnik, Statistical learning theory, Vol. 1, Wiley New York, 1998.

[36] C. Tzelepis, V. Mezaris, I. Patras, Linear maximum margin classifier for learning from uncertain data, IEEE transactions on pattern analysis and machine intelligence 40 (12) (2018) 2948–2962.

[37] V. Mygdalis, A. Tefas, I. Pitas, Exploiting multiplex data relationships in support vector machines, Pattern Recognition 85 (2019) 70–77.

[38] J. Weston, C. Watkins, Multi-class support vector machines, Tech. rep., Citeseer (1998).

[39] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE transactions on Neural Networks 13 (2) (2002) 415–425.

[40] U. Doğan, T. Glasmachers, C. Igel, A unified view on multi-class support vector classification, Journal of Machine Learning Research 17 (45) (2016) 1–32.
URL http://jmlr.org/papers/v17/11-229.html

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[42] J. S. Bridle, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, in: Neurocomputing, Springer, 1990, pp. 227–236.

[43] Y. Tang, Deep learning using linear support vector machines, arXiv preprint arXiv:1306.0239.

[44] Y.-J. Lee, O. L. Mangasarian, Ssvm: A smooth support vector machine for classification, Computational optimization and Applications 20 (1) (2001) 5–22.

[45] K. Janocha, W. M. Czarnecki, On loss functions for deep neural networks in classification, arXiv preprint arXiv:1702.05659.

[46] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.

[47] C. E. Shannon, A mathematical theory of communication, ACM SIGMOBILE mobile computing and communications review 5 (1) (2001) 3–55.

[48] D. W. Scott, Multivariate density estimation: theory, practice, and visualization, John Wiley & Sons, 2015.

[49] S.-T. Chiu, Bandwidth selection for kernel density estimation, The Annals of Statistics (1991) 1883–1905.

[50] M. Koestinger, P. Wohlhart, P. M. Roth, H. Bischof, Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 2144–2151.

[51] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: European Conference on Computer Vision, Springer, 2014, pp. 94–108.

[52] S. Yang, P. Luo, C.-C. Loy, X. Tang, Wider face: A face detection benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.

[53] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS workshop on deep learning and unsupervised feature learning, Vol. 2011, 2011, p. 5.

[54] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images.

[55] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.

[56] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.