Detection of the Presence of Safety Helmets on Motorcyclists Using Active Appearance Models

Felipe José Aguiar Maia, José Everardo Bessa Maia, Thelmo Pontes de Araujo
State University of Ceará (UECE)
Brazil
felipe.ja.maia@gmail.com
jose.maia@uece.br
thelmo.araujo@uece.br

ABSTRACT: *This work looks into the problem of detecting the presence of motorcyclist safety helmet on single images of city roads. The degree of adjustment of a new image to the previously trained Active Appearance Model (AAM) acts as the decision criterion for detection. This adjustment is measured by differences between AAM shape and appearance parameter vectors, which become the feature vectors for four different classifiers. Results show that this approach is feasible and may be recommended.*

Keywords: Safety Helmet Detection, Aam Features, Pattern Recognition

Received: 25 April 2018, Revised 12 June 2018, Accepted 19 June 2018

DOI: 10.6025/jic/2018/9/4/157-165

1. Introduction

Algorithms to solve the problem of detecting multiples instances of an object in a given image and verifying some property in those instances are still a challenge, specially in very general scenarios. Variations on lighting, object silhouette, and background are some of the difficulties usually found. This short paper aims to detect motorcyclists on single images of city roads and to determine the presence of safety helmets.

Images of city roads obtained by transit monitoring systems usually have poor quality, due to lighting variations, rain, etc. For this reason, Active Appearance Model (AAM) features were chosen as features for classification, since they work well with ill-defined shapes [5].

The problem of locating and tracking motorcyclists on videos of public roads is approached by many authors [13, 7, 1]. [17] use Haar features, histograms of gradient images, and Local Binary Patterns (LBP) as features to be classified by classifiers such as

Radial Base Function neural network (RBF), Multi Layer Perceptron neural network (MLP), and Support Vector Machine (SVM). Background is suppressed to reduce processing. That work was extended [16] to determine the presence of safety helmets on the image most probable regions for the motorcyclist's head.

In [12], a search for circle-like contours is made in order to detect safety helmets in located motorcyclists. The presence of occlusions is considered. [4] locates moving vehicles after removing the image background. Then, motorcycles and cars are classified based on their proportions. Histograms of the most probable regions for the motorcyclist's head are used to determine the presence of a safety helmet.

The use of deep neural networks has increased in the last decade. [15, 19, 20] used Convolutional Neural Networks (CNN) to detect a motorcyclist in an image and to decide whether he is wearing a helmet or not.

Most papers in this area [12, 17, 16, 4] have video stream as data instead of single images. However, using a single image is necessary when the helmet verification follows another detection process such as a red traffic light violation, which is precisely our case. In fact, the use of a sole image is a more challenging task, since video streams contain information on inter-frame temporal variations, which may be used for feature extraction in the classification process. In this work, we evaluate shape and appearance parameter vectors obtained from an AAM as feature vectors [11] to verify the presence of safety helmet in motorcyclists using four supervised classifiers. AAM were also successfully used in other tasks closely related to this work, such as facial recognition [14], segmentation [3], and pose estimation [2].

"Presence of safety helmet" is chosen as the goal class for the artificial shape and texture of helmets are more distinguishable on general scenes than human heads without a helmet.

This paper is structured as follows: Section 2 shows how AAM vectors are obtained. Section 3 describes the methodology. Results and discussion are in Section 4. Section 5 concludes the paper.

2. Active Appearance Models

Active Appearance Models (AAM) were proposed by [5] as an extension to Active Shape Model (ASM) [6] and are able to model both shape and appearance. Shape is understood as a set of ordered labeled landmarks intended to capture the spatial form of an object (and also its parts). By its turn, appearance is the texture of an object represented by the gray level intensity (or color) of its pixels on an image.

The first part of AAM and ASM algorithms is the creation of the Point Distribution Model (PDM), consisting of optimizing the alignment (by translation, rotation, and scale) of the original landmark points on the training set with a mean shape (Figure 1).

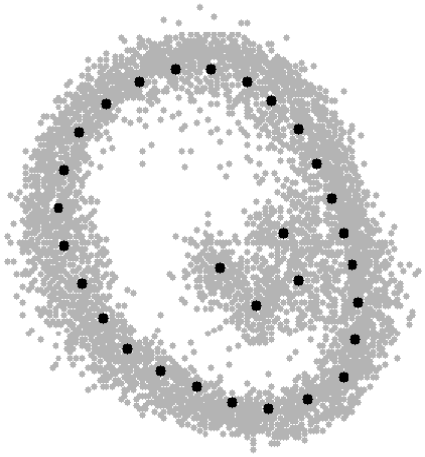

Figure 1. Mean shape (black dots) and aligned landmarks (gray dots)

Principal Component Analysis (PCA) [10] is then applied to the training data to obtain an approximation of the shape:

$$x = \bar{x} + P_s b_s, \tag{1}$$

where $\bar{x}$ is the mean shape, P_s is the matrix with the main orthonormal eigenvectors of the (shape) data covariance matrix, and b_s is the vector which describes shape x.

Once the mean shape is computed, a triangulation algorithm (such as Delaunay's) is applied to it (Figure 2, middle), with the landmarks being the vertices of the triangles. This same triangulation is reproduced on each landmark labeled training images (Figure 2, left). Finally, the pixels inside each triangular region on each training image are mapped into the corresponding triangle on the mean shape image, creating, for each training image, a distorted image that matches the mean shape region (Figure 2, right).

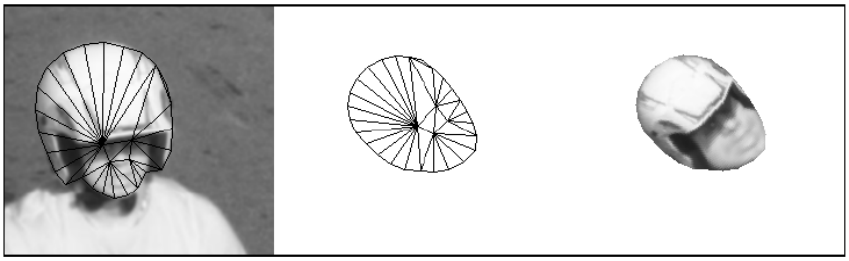

Figure 2. Image mapped to the mean shape region via deformation of the triangulation

The pixels inside each mapped image are normalized to reduce lighting variability and then vectorized. PCA is applied to those vectors to obtain a linear model for the texture:

$$g = \bar{g} + P_g b_g; \tag{2}$$

where $\bar{g}$ is the normalized average of gray level vectors in the training set, P_g is the matrix with the main orthonormal eigen vectors of the (texture) data covariance matrix, and b_g is the vector which describes appearance g.

Vectors b_s and b_g describe, respectively, the shape and the appearance of each original image. To account for any correlation between them, PCA is applied a third time to the vectors

$$b = \begin{bmatrix} W_s b_s \\ b_g \end{bmatrix} = \begin{bmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{bmatrix}, \tag{3}$$

where the diagonal matrix W_s compensates the difference between shape and texture units, obtaining

$$b = Qc, \quad \text{with } Q = \begin{bmatrix} Q_s \\ Q_g \end{bmatrix}, \tag{4}$$

where Q has orthonormal columns formed by the main eigenvectors and c describes both shape and appearance. Because the model is linear, shape and appearance may be described separately by c using:

$$x = \bar{x} + P_s W_s Q_s c \quad \text{and} \quad g = \bar{g} + P_g Q_g c. \tag{5}$$

Given a new image and the appearance model previously described, the model parameters may be adjusted in such way that the synthesized image matches the new image the best as it can. This may be treated as an optimization problem in which the squared norm of the difference between the new image and the synthesized one must be minimized:

$$\min \|\delta I\|^2, \quad \text{with } \delta I = I_i - I_m, \tag{6}$$

where I_i is the new image gray levels and I_m is the gray levels synthesized by the appearance model.

Ideally, the minimum value on Equation (6) is obtained by varying the parameter vector c (corresponding to I_m) on Equations (5), but this procedure reveals to be too expensive. Hence, we use regression to find a linear relation between the variation on the parameters (δc) and δI:

$$(\delta c) = A\delta I \qquad (7)$$

Synthetic images may be generated (using Equations (5)) by perturbing the parameter vector δc of images on the training set. Small perturbations on scale and pose may also be added. Matrix A is computed according to the following steps:

• From an original training image, c_0 may be determined and then perturbed by a given c to get $\delta c = c_0 + \delta c$;

• Shape and appearance vectors x and g_m may be computed for the perturbed vector c;

• The original image is deformed to match its shape with the shape vector x and then its normalized appearance vector g_s is computed;

• The appearance difference is calculated by $\delta g = g_s - g_m$;

• Linear regression applied to $\delta c = A\delta g$ gives matrix A.

Now, the process may be run for a new image outside the training set. Given a new image, a reasonable region of interest (ROI) is located and the PDM parameter vectors $\bar{x}$ e $\bar{g}$ are scaled accordingly. Algorithm 1 is then applied.

Data: PDM parameters: $\bar{x}$, $\bar{g}$, P_s, W_s, Q_s, P_g, Q_s; A; new image appearance vector g_s.

Result: Final shape, appearance vectors: x, g.

Initialize c_0, $g_0 = g$, $\Delta\in$, $\Delta\in_{max}$;

while $\Delta\in > \Delta\in_{max}$ do

    $\delta g = g_s - g_0$; $\in_0 = ||\delta g||^2$; $gc = A\delta g$;

    $\eta = 2$;

    while $\in > \in_0$ or $\eta > 0{:}001$ do

        $\eta = \eta/2$;

        $c = c_0 - \eta\delta c$;

        $g = \bar{g} + P_g Q_g c$;

        $\delta g = g - g_s$;

        $\in = ||\delta g||^2$;

    end

    update: $g_0 = g$; $c_0 = c$; $\Delta\in = |\in - \in_0|$; g_s;

end

compute: $x = \bar{x} + P_s W_s Q_s c$; $g = \bar{g} + P_g Q_g c$

Algorithm 1. Adjusting the model to a new image

3. Methodology

In this paper, the methodology is comprised of three phases: detection of the ROI, feature extraction, and classification (Figure 3).

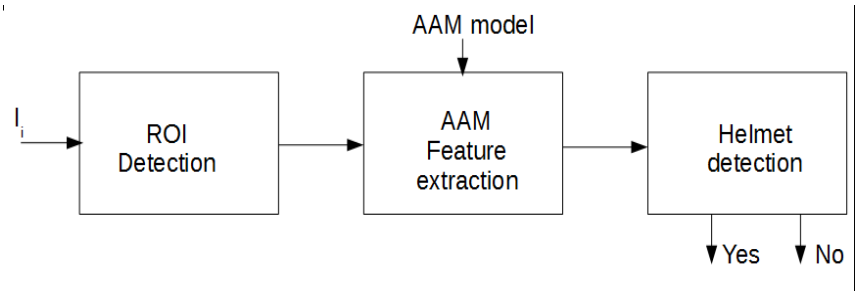

Figure 3. Procedure flow diagram

Unlike works that use video streams [12, 17, 16, 4], we use a sole static image to determine the presence of safety helmet, a constraint due to the image acquirement process—a picture taken after a red traffic light violation. In order to detect the presence of motorcyclists in the experiments on this paper, the Viola-Jones algorithm—based on Haar features and AdaBoost —was chosen for its generalization power and robustness [18]. The algorithm was trained until a reasonable false negative rate was achieved and then applied to the images in the training set. Multiple ROI detection on a single image still occurred.

Figure 4 shows positive detection examples of the ROI: (a) shows an example of a single ROI detection of a motorcyclist without the safety helmet; (b) shows an example of a double ROI detection of a motorcyclist with the safety helmet.

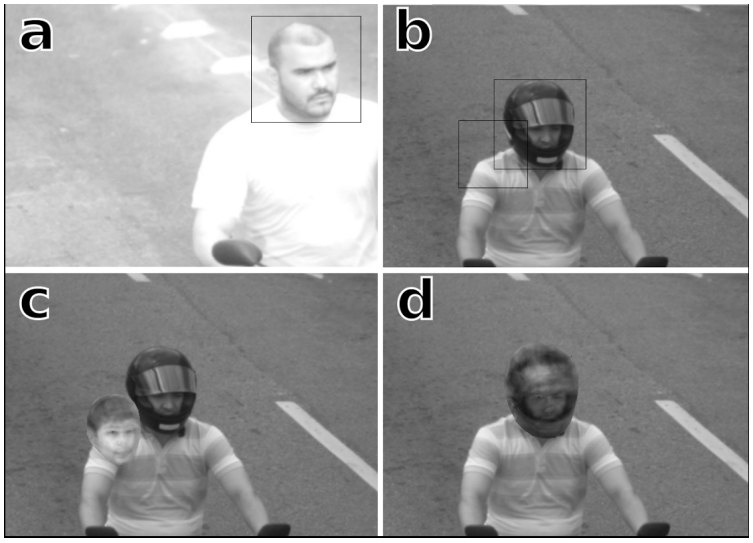

Figure 4. ROI and positive head detections (single and double)

The ROI detection has to be executed with a good precision, because the displacements computed in the AAM algorithm have to be small enough to guarantee the linearity of the relation between error and adjustment, which is necessary to the correct use of linear regression [5]. In the case of not finding a motorcyclist, the execution is aborted. In order to discard false positives in double detection cases (like the one seen in Figure 4 (b)), we select the parameter vector closer to its corresponding mean model vector, thus eliminating the undesirable detection (Figure 4 (c)(d)).

Once the ROI was detected, the AAM algorithm described in Section 2 is executed in order to extract the image features. Figure 5 shows an example of the AAM algorithm convergence for the case with safety helmet: the original image is on the left; the model approximation (using only a few features, $n = 206$) is shown on the right.

Figure 5. AAM convergence: case with helmet

For the classification phase, four classifiers were compared [8]: Support Vector Machine (SVM) with linear kernel, SVM with radial basis kernel, SVM with non-linear polynomial kernel, and Multi-Layer Perceptron (MLP) neural network.

Since the basic SVM performes well only when data are linearly separable, non-linear kernels must be used in order to map the data into a high dimensional feature space in which linear separation may be achieved [9]. Here, besides the linear kernel, a non-linear polynomial kernel:

$$K(x, y) = (\gamma x^T y + c)^d \,, \quad \text{with } \gamma > 0 \,,$$

and a radial basis kernel:

$$K(x, y) = \exp\left(-\frac{|x - y|^2}{\sigma^2}\right)$$

were used, where σ is the scale factor.

In the following section, results show the performance of each classifier considering the feature vectors extracted by the AAM algorithm, and the output is either the class "with helmet" or the class "without helmet".

4. Results

4.1 Experiments

All computational implementations used C++ language and the computer vision library OpenCV. The dataset is composed by 1060 images, 530 of each class, from which 800 images (400 of each class) are used to train the Viola-Jones algorithm and 260 (130 of each class) for training and testing the classifiers.

We modified only the following parameters in the OpenCV's Viola-Jones algorithm: false positive maximum rate of 0.01; 8 training stages; maximum depth of 4 layers; maximum counting of weak classifiers of 5; window size of 24×24. The cascade use Haar features.

A 10-fold cross validation scheme were used to train and test the classifiers, using 260 images. All images were manually labeled with 30 landmarks. To generate the PDMs, we set the PCA to keep 99% of variability, which means that the number of parameters is variable.

The regression matrix that approximates the model used 50 random perturbations of each example in the training set, with +2 times the standard deviation.

For the SVM classifiers, we used the OpenCV implementation with penalty factor $C = 0.1$. RBF kernel is a Gaussian with scale

factor $\sigma = 10^{-5}$. The polynomial kernel is $K(x, y) = (\gamma x^T y + c)^d$, where $\gamma = 10^{-5}$, $c = 274.4$, and $d = 0.07$. As for the MLP, we used a single hidden layer topology, with twice the number of parameters as neurons, learning rate of 0.1, and momentum of 0.1.

4.2 Results and Discussion

After applying the Viola-Jones algorithm, we selected the true positive detected images and proceeded with the classification stage, whose results are shown in Table 1. Table 1 shows the basic measures from the confusion matrix, i.e., True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), and False Negative Rate (FNR); as well as the derived measures Accuracy, Area Under the ROC (AUC), Precision (or Positive Predictive Value – PPV), Negative Predictive Value (NPV), and F-measure (or F_1 score).

As can be seen on Table 1, SVM with linear kernel gives the second worst mean accuracy (71:7%), despite achieving the best (lowest) false negative rate (FNR = 5:83%). As an automated system to support traffic law enforcement, a low FNR is desirable (otherwise fines could be unfairly applied), but a false positive rate (FPR) of 54:3% is not generally acceptable.

The SVM with RBF kernel had the best (higher) TNR (87:2%)—and the best (lower) FPR (12:8%)—while having the worst (higher) FNR (= 29:1%). This could help traffic authorities to punish motorcyclists that fail to wear the safety helmet (a good thing), but increase the need for human verification in the system, in order to avoid unfair traffic tickets.

The MLP classifier results were similar to those of the SVM with linear kernel, being a good option to the latter when a low FPR were more desirable than a low FNR.

The relevance of metrics on Table 1 depends on the application goal: If the goal is to detect motorcyclists wearing safety helmets in an area where facial identification is necessary for security reasons, TPR is the most relevant metric. SVM with linear kernel classifier achieved the best results with *TPR* = 94:2%. In this case, FNR is not so relevant, since asking someone without a helmet to take it off is not very troublesome. If the application aims to detect motorcyclists without the safety helmet in order to punish them, then TNR is the most relevant metric. In this case, SVM with RBF kernel performed better, with TNR = 87:2%. However, its FNR is the highest (worst) one (= 29:1%), and it would be necessary to check whether the costs of this type of error are acceptable for the application.

In terms of mean accuracy, the performance of SVM with non-linear polynomial kernel and MLP classifiers are practically the same, with a little advantage for the SVM with non-linear polynomial kernel classifier (mean accuracy = 81:2%). All results for SVM with non-linear polynomial kernel were good and well balanced, making this classifier the best general choice among the four tested ones.

	Classifier			
	SVM Linear	SVM Polynomial	SVM RBF	MLP
TPR (Recall)	0.942	0.884	0.709	0.913
FNR	0.058	0.117	0.291	0.087
TNR	0.457	0.734	0.872	0.681
FPR	0.543	0.266	0.127	0.319
Precision	0.634	0.769	0.848	0.741
NPV	0.887	0.863	0.750	0.887
F-measure	0.758	0.822	0.772	0.818
AUC	0.862	**0.879**	0.794	0.849
Accuracy μ (σ^2)	0.717 (0.0095)	**0.812** (0.0125)	0.680 (0.0032)	0.802 (0.0068)

Table 1. Performance results for the four classifiers

ROC curves, on Figure 6, allow us a better evaluation of the trade-off between the two types of error in all four classifiers. They also allow, for a given FPR, to estimate the TPR of each classifier. For example, if an FPR = 30% is admissible for the application, ROC curves show that the SVM with polynomial kernel classifier achieves an TPR close to 90%.

The values of the area under the ROC curve (AUC) show very similar results for three classifiers (SVM with linear kernel, SVM with non-linear polynomial kernel, and MLP), so AUC should not be used as the sole classification performance criterion.

Figure 6 also confirms that the SVM with non-linear polynomial kernel and the MLP classifiers were the best ones among the four classifiers used in our comparison. Moreover, they achieved very similar results.

The derived measures help to select the best classifier for a given application: SVM with RBF kernel classifier performs better when Precision is important; and both SVM with linear kernel and MLP classifiers are prefferable when NPV is important. However, SVM with non-linear polynomial kernel has the most balanced performance, since it has the highest accuracy and F-measure.

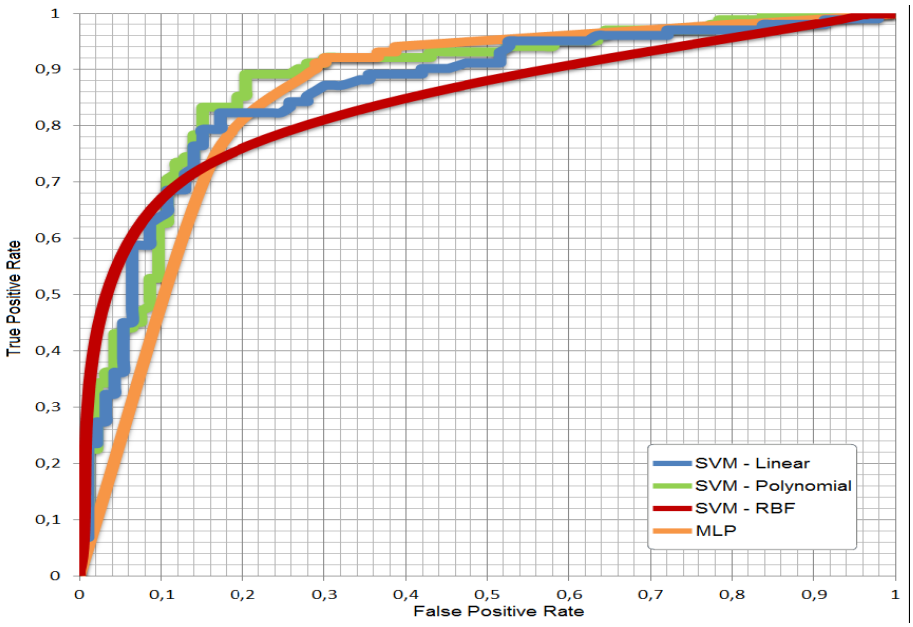

Figure 6. Location of classifiers in ROC space

5. Conclusion

AAM features were evaluated to detect the presence of safety helmets on motorcyclists in single images of city roads, which is a common and more challenging task than using video streams for classification. As far as the authors know, no results with similar settings were found in the literature for comparison. One may notice that the test results obtained here with single images were only slightly inferior (mean accuracy = 81:2%) to the ones on previous work based on video stream data ([4], mean accuracy = 85%), although those have much more information. Therefore, our approach reveals to be promising, since video stream data are more complex and more computationally expensive.

In summary, SVM with non-linear polynomial kernel achieved a more balanced performance in comparison with the other classifiers evaluated here, as can be seen in the mean accuracy, AUC, and F-measure results. When the relevant issue is safety helmet absence, the results for NPV point to SVM with linear kernel and for MLP as the best classifiers; on the other hand, when the presence of the safety helmet is more important, Precision results indicate SVM with RBF kernel as the best classifier.

This research may continue with the use of AAM based on color tensors as feature vectors for classification.

Acknowledgments

[double-blind] acknowledges the financial support from [double-blind].

References

[1] Ashvini, M., Revathi, G., Yogameena, B., Saravanaperumaal. S. (2017). View invariant motorcycle detection for helmet wear analysis in intelligent traffic surveillance. *In*: Proceedings of International Conference on Computer Vision and Image Processing, p 175–185. Springer, 2017.

[2] Navid Mahmoudian Bidgoli., Abolghasem A Raie., Naraghi, M. (2014). Probabilistic principal component analysis for texture modelling of adaptive active appearance models and its application for head pose estimation. *IET Computer Vision*, 9(1):51–62, 2014.

[3] Ruida Cheng., Holger R Roth., Le Lu., ShijunWang., Baris Turkbey., William Gandler., Evan S McCreedy., Harsh K Agarwal., Peter Choyke, Ronald M Summers, et al. (2016). Active appearance model and deep learning for more accurate prostate segmentation on mri. In Medical Imaging 2016: *Image Processing*, volume 9784, p 97842I. International Society for Optics and Photonics 2016.