IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 12, 2020, accepted December 30, 2020, date of publication January 11, 2021, date of current version January 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050514

Learning to See Through a Few Pixels: Multi Streams Network for Extreme Low-Resolution Action Recognition

PAOLO RUSSO, SALVATORE TICCA, EDOARDO ALATI, (Member, IEEE), AND FIORA PIRRI, (Member, IEEE)

Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti, University of Rome La Sapienza, 00100 Rome, Italy

Corresponding author: Paolo Russo (prusso@diag.uniroma1.it)

This work was supported by the Horizon 2020 EU Research and Innovation programme, project SECONDHANDS, under Grant 643950.

ABSTRACT Human action recognition is one of the most pressing questions in societal emergencies of any kind. Technology is helping to solve such problems at the cost of stealing human privacy. Several approaches have considered the relevance of privacy in the pervasive process of observing people. New algorithms have been proposed to deal with low-resolution images hiding people identity. However, many of these methods do not consider that social security asks for real-time solutions: active cameras require flexible distributed systems in sensible areas as airports, hospitals, stations, squares and roads. To conjugate both human privacy and real-time supervision, we propose a novel deep architecture, the *Multi Streams Network*. This model works in real-time and performs action recognition on extremely low-resolution videos, exploiting three sources of information: RGB images, optical flow and slack mask data. Experiments on two datasets show that our architecture improves the recognition accuracy compared to the two-streams approach and ensure real-time execution on Edge TPU (Tensor Processing Unit).

INDEX TERMS Action recognition, activity recognition, deep learning, computer vision, multi-modal, low resolution.

I. INTRODUCTION

Video acquisition technologies are becoming pervasive in our daily lives. Powerful digital cameras used in social media, traffic, security and emergency monitoring are capable of capturing high-level details of people's face and body, causing severe privacy issues [3], [8]. Using data acquired by these ubiquitous cameras, deep architectures have investigated and faced privacy issues, especially regarding activity recognition, in [17], [18]. However, these approaches model deep architectures requiring high computational power, disregarding real-time performance and integration into embedded devices. As a matter of fact, efficiency and privacy are not jointly faced, so far.

To face these drawbacks we propose to work on both *Low Resolution* (LR) and *extremely Low Resolution* (eLR) data, using light deep architectures. Our approach's advantage is two-fold: it keeps relevant information for security and social issues, avoiding sensible data acquisition and at the same time it allows real-time recognition and implementation on digital devices [1]. See both Figure1 and Figure 3.

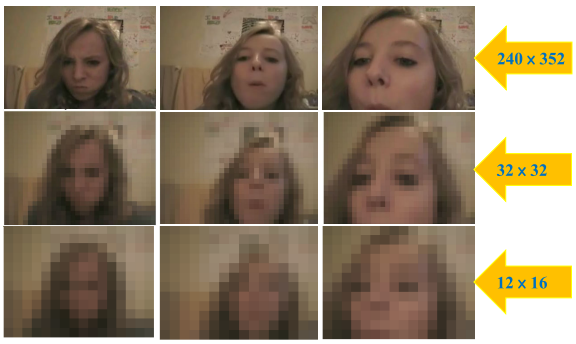


FIGURE 1. Privacy preservation. The Figure shows how very low resolution (12×16) suppresses face details, while some details are still visible with 32×32 resolution. Images taken from HMDB51 videos dataset.

Our novel architecture, the *Multi Stream Network (MSN)*, can perform activity recognition tasks on eLR data by exploiting **three sources** of information: RGB, optical flow, and slack foreground masks. These three streams are processed in

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao.

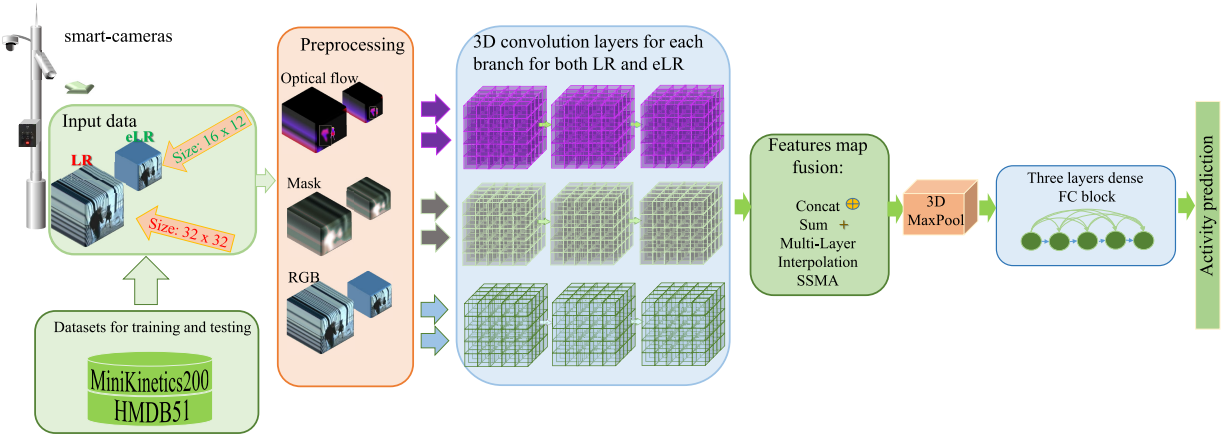


FIGURE 2. The MSN architecture. The deep model takes both LR and eLR video-data for training. For inference, it can take only eLR video-data, being able to acquire videos from smart-lamppost with edge-computing facilities. The input video-data are preprocessed along three branches: optical flow, taking care of time and motion, slack mask, taking care of the foreground shapes, and spatial features via the RGB channels. Preprocessed data, along each branch, are passed to $5\times5\times5$ convolutional 3D layers, composed respectively of 32, 64 and 128 filters. The feature maps obtained by the 3D convolution branches are fused, according to methods described in Paragraph III-B. Finally, the fused features map is elaborated by a three-layer, fully connected dense network leading, via softmax, to activities prediction.

FIGURE 3. Example of a frames sequence from the HMDB51 dataset. The first row shows the RGB sequence at the original resolution. The second row presents the downsampled LR frames, while the third one contains the same sequence's optical flow. Here, we can see the camera motion combined with both the horse-rider and the water motion. Finally, the fourth row shows the downsampled sequence slack mask.

separate 3*D* convolutional branches, which are then fused for the final classification (see Figure 2). Detailed experiments on an eLR versions of HMDB51 [12] and on Mini-Kinetics 200 [27], validate the power and time performance of the proposed approach against the methods using just RGB and optical flow.

II. RELATED WORKS

Action recognition has been deeply studied in the last years. Simonyan *et Al.* [19] proposed a two-stream CNN network to incorporate visual and motion features: one stream processes RGB frames while the other processes the optical flow. Several works exploit the multi-source paradigm, such as [2], [23] integrating also different kinds of information, from multi-region focus [31] to multiple sensors [20]. Convolutional 3D (C3D), introduced in [21], bypassed the need of the optical flow relying on 3D convolutional layers to learn high informative spatial-temporal features. Inception3D (I3D) [5] improved the aforementioned approaches proposing a novel Two-Stream Inflated

3D ConvNet. Thereafter, Residual(2+1)D (R(2+1)D) [22] introduced a new decomposed convolutional network, which separates spatial-temporal modelling, cutting down the number of parameters while maintaining comparable performances.

The eLR action recognition has been tackled by a number of methods focusing either on high accuracy or on light-weight architectures. While both approaches can remove privacy details, the light-weight ones are less accurate, though real-rime oriented, not resorting to heavy spatial-temporal models such as I3D or R(2+1)D. We refer to the two approaches as the *heavy* and *light* approaches. We place our approach among the *light* ones.

Among the *heavy* approaches, Purwanto *et Al.* [15] introduce a novel spatial-temporal multi-head self-attention mechanism based on the combination of super-resolution, knowledge distillation and attention mechanisms to learn powerful yet general features. Moreover, Purwanto *et Al.*, in [14], address the visual degradation problem with an additional input source, namely the trajectory. This architecture complements the usual spatial-temporal information with trajectory patterns capturing features robust to visual distortion. The work of Xu *et Al.* [29] can also be placed in the *heavy* set, as it introduces a fully-coupled architecture, which incorporates 3D CNN and RNN to extract enhanced and more robust video motion representation. Even if still working with *eLR* images, the approaches mentioned above are essentially built on top of I3D pre-trained on Kinetics [10], exploiting a massive number of parameters via an up-sampling step. These methods improve the accuracy at the expense of a higher computational cost. Ryoo *et Al.* [17] approach can be considered *heavy* too, since it exploits high resolution videos. It decomposes high-resolution training videos into multiple low-resolution ones and uses a multi-siamese model to learn a shared spatial embedded space.

In the *light* set, we place both [6] and [18], which we compare with in Section IV. Chen *et Al.* [6] introduce a semi-coupled two-stream network exploiting a partial sharing of the weights to benefit from high resolution videos in training. Ryoo *et Al.* [18] propose a method that relies on learning inverse super-resolution based on Markov Chain Monte Carlo (MCMC), and using Histogram of Oriented Gradients (HOG) features. We show that MSN outperforms both [6] and [18].

The great advantage of the *light* methods is the possibility to work efficiently in distributed architectures, working directly on Edge TPU-equipped remote devices, on the base of the performance benchmarked by [25], [26].

III. METHOD

MSN is a novel multibranch convolutional network, where each branch performs a pipeline of $3D$ convolutional layers on a different input source. A sequence of K frames is taken as input: the first branch computes optical flow, the second a slack foreground mask for each frame, and the last branch computes spatial features along the RGB channels. Every branch produces a features map; these are fused to feed a fully connected (FC) dense three hidden layers network for the activity classification. The training loss is a standard categorical cross-entropy. We experimented several types of features fusion approaches: a brief description is reported in Paragraph III-B.

Our method attacks the problem of computational cost via a trade-off between features extraction, pre-processing and convolutional $3D$ network computation efficiency. We adopt a hybrid method to extract information at different levels: training is performed by alternating *eLR* and *LR* data, respectively having 16×12 and 32×32 resolution. This methodology enables features transfer learning from *LR* to *eLR* data [6]. Nevertheless, at inference time, only *eLR* data is used to perform activity recognition.

All in all, our method differs from the state of the art algorithms in several aspects:

- It exploits a *light* model with the number of parameters decreased by an order of magnitude with respect to traditional, burdensome pre-trained architectures (see Table 1).
- The 3D convolutional network improves the approaches based on the combination of 2D convolutions and mean over predictions (see [6]), with a sensible boost in performance.
- MSN uses the slack mask stream to add further features information to the other two streams bringing an additional improvement.

A. ARCHITECTURE DETAILS

Each MSN branch is made of three convolutional 3D layers of dimension $5\times5\times5$ composed respectively of 32, 64 and 128 filters, interspersed with Batch Normalization and Relu non-linearities. Despite its large size compared to the input resolution, this particular kernel proved to give the best accuracy, as can be seen in Section (V), Table 7. Each branch's feature maps are fused and the resulting map, following a 3D max-pooling, is fed to a fully connected neural network composed of three hidden layers, with dropout.

B. FEATURES FUSION METHODS

MSN can be modelled with several feature combination approaches. These methods have been tested with each of the feature maps F_{RGB}, F_{OF} and F_M computed, respectively, from the RGB, optical flow (OF) and mask (M) streams. Each feature map can be defined as $F \in \mathbb{R}^{(nK\times ds)}$, with nK the number of kernels, K the number of considered frames, and $ds = n\times n\times n$ the downscaling tensor. A discussion on the tested methods is then proposed.

Concat: concatenates the output feature maps of every branch along the channel dimension:

$$y_{cat} = F_{RGB} \oplus F_{OF} \oplus F_M. \tag{1}$$

With $\oplus$ the concatenation symbol. This method has the advantage of keeping the feature maps extracted from each branch.

TABLE 1. The Above Table Compares the Number of Parameters and the Frame Rate of I3D and MSN, Along the Different Fusion Methods we Adopted, as Described in Paragraph III-B.

Architecture	I3D	MSN + Sum	MSN + Concat	MSN + Interpolation	MSN + SSMA	MSN + Multi-Layer
# Parameters	$\sim 25M$	**4.557.472**	5.606.084	4.999.840	5.755.080	5.897.472
Frame Rate	$\sim 8fps$	$\sim 333fps$	$\sim 315fps$	$\sim 300fps$	$\sim 280fps$	$\sim 290fps$

Sum: performs the features element-wise addition:

$$y_{sum} = F_{RGB} + F_{OF} + F_M. \tag{2}$$

Unlike the previous method, the **Sum** procedure main advantage is to build a fused vector of the same size of the single-stream output F_X, with $X \in \{RGB, OF, M\}$, in so avoiding to increase the number of parameters as done by the concatenation method. Moreover, concerning concatenation, the **Sum** method reduces the model overfitting.

Multi Layer: is inspired by the early fusion approach proposed in [16], which combines the features at multiple scales. This method allows the three streams interaction from the early stages. For example, after the i-th convolutional layer, the streams are fused in the following way:

$$\begin{aligned} \hat{y}_{ml}^{(i)} &= F_{RGB}^{(i)} + F_{OF}^{(i)} + F_M^{(i)} + y_{RL}^{(i-1)} \\ y_{RL}^{(i)} &= ReLU(\hat{y}_{ml}^{(i)} * w^{(i)} \kappa_{1\times1\times1}). \end{aligned} \tag{3}$$

Here $F_{RGB}^{(i)}$, $F_{OF}^{(i)}$ and $F_M^{(i)}$ are the feature maps computed by the i-th layer of the RGB, optical flow and mask streams, respectively, and $y_{RL}^{(i-1)}$ is the outcome of their fusion at the previous layer, with $w^{(i)}$ denoting the weights of a $(1\times1\times1)$ 3D convolutional kernel κ, $i \geq 0$, with $\hat{y}_{ml}^{(0)}$ as y_{sum} in eq. (2).

Interpolation: loosely inspired by [15], computes a weighted sum of the feature maps by applying different weights to each stream. In this way, the most informative features gain more importance:

$$\begin{aligned} y_{inter} &= w_{RGB} \odot F_{RGB} + w_{OF} \odot F_{OF} + w_M \odot F_M, \\ w_a &= \sigma((y_{cat} * \kappa_I) \oplus F_a) * \kappa_{s_a}), a \in \{RGB, OF, M\}. \end{aligned} \tag{4}$$

The weights $w_a \in \mathbb{R}^{(nK\times ds)}$ are computed as a function of the convolutional kernel κ_I (*stream independent*) and κ_{s_a} (*stream specific*), which are jointly learned during the MSN training. The symbol $\odot$ indicates element-wise product between matrices.

Self-Supervised Model Adaptation (SSMA) [24] adaptively fuses the three branches using a novel reduced block tailored to work at low resolutions:

$$y_{ssma} = y_{cat} \odot \sigma(y_{cat} * w_k \kappa_{1\times1\times1}) \tag{5}$$

Here w_k denotes the weight of a $(1 \times 1 \times 1)$ 3D convolutional kernel $\kappa_{1\times1\times1}$.

IV. EXPERIMENTAL SETUP

MSN has been extensively tested on HMDB51, a popular action recognition dataset. It has been tested on an eLR version of Mini-Kinetics 200 too. In the following, we introduce details on the datasets used, on the preprocessing generating the three streams input to the three MSN branches, and on the MSN network hyperparameters.

A. DATASETS

HMDB51 [12] is one of the most widespread benchmark for activity recognition. It comprises movie clips and YouTube videos at a different resolution for 51 action categories with at least 101 videos per class, for 6849 videos. The evaluation procedure consists of averaging accuracy obtained on the three train/test splits provided by the dataset authors.

Kinetics [28] is a large-scale YouTube video dataset showing human actions. These cover a broad range of human activities up to 700 classes and 650K video clips.

Mini-Kinetics 200 is a subset proposed by [27]. It comprises 200 of the Kinetics classes containing the highest amount of samples. MSN has been tested on an eLR version of this dataset by performing a downscale of each frames sequence.

B. PREPROCESSING

Video preprocessing is required to keep the deep architecture tight and agile, and it is computed on downscaled frames to a size of 32×32 to obtain the *LR*. We use the same standardization introduced by [6] to perform a fair comparison.

The first preprocessing stream is the optical flow, computed according to the dualTV-L_1 method of [30]. Optical flow generates features of the scene about the camera and people motion, tracking the motion along time. The OpenCV implementation is quite fast for LR videos, consuming about 2 *ms* for each pair of frames on CPU. Although the camera motion is combined with scene motion in most videos, there are static videos in which the optical flow contribution is scarce. Therefore the slack mask is beneficial in particular in these cases.

The second stream is the slack mask computed as follows. Let V be an input video of length N, with frames $f_1, \ldots, f_N$, and let G be a Gaussian kernel of size 3×3 with $\mu = 0$ and $\sigma < 1$. We first smooth each frame by convolving it with G, namely, $f_i^* = f_i * G$, for $i \in \{1, \ldots, N\}$, with $*$ the convolution operator, and rescale it to 32×32, padding it to fit to the original image ratio. Further, we convolve f_i^* with a Sobel kernel of size 5, in so obtaining f_i^{*S}. Using the transformed frames f_i^{*S} we generate two similar videos V_1^* and V_2^*, just shifted by one frame, namely, V_1^* is $(f_1^{*S}, \ldots, f_{N-1}^{*S})$ and V_2^* is $(f_2^{*S}, \ldots, f_N^{*S})$. Then we compute:

$$A_n = \frac{1}{k} \sum_{n}^{n+k} g_n^{*S} * G, \text{ for } g_n^{*S} \in \{\|V_1^\star - V_2^\star\|\}_{n=1}^{N-(k-1)}. \tag{6}$$

Here $\|V_1^\star - V_2^\star\|$ is the absolute difference between the two frame sequences V_1^* and V_2^*, and $*$ is the convolution operator. A might require an extension with up to k frames

according to the modulo ($N \mod k$). If $(1/N)\sum_n A_n > \tau$, with τ a threshold fixed to 0.1, then the final mask for M_n of frame n is the point wise matrix product between A_n and the Laplace transform of frame f_n^*:

$$M_n = A_n \odot \nabla \cdot \nabla f_n^*. \tag{7}$$

Otherwise, we take simply the Laplace transform of $f_n^\star$, computing it with a 3×3 kernel. Here $\odot$ is the element-wise matrix product, and we considered $k = 3$, the size of the Gaussian kernel. To understand the above formulation's simple idea, consider when motion is in the image, then the transformations leading to M_n highlight regions with the largest gradient, namely in the foreground, and the Laplace operator further enhance them. On the other hand, if $(1/N)\sum_n A_n$ is less than τ then the video has virtually no motion; therefore the Laplace operator, on the low-resolution Gaussian smoothed frames, is enough to return a flat region on the foreground showing large contours around it. The obtained results are shown in Figure 3.

C. NETWORK HYPERPARAMETERS

After the preprocessing step, the video frames are downscaled to 16×12 to generate the eLR data and then upsampled again to 32×32, matching the network working resolution.

We use a zero-mean Gaussian distribution to initialize the network weights, assumed i.i.d, with a standard deviation proportional to the square root of 2 over the number of layer connections, as suggested in [9]. The training have been performed with the *Adam* Optimization algorithm [11]. The learning rate tuned on grid search [4], [7] starts at a value of 10^{-4} and it is scheduled to decrease up to 10^{-6}.

To limit overfitting, we regularize via a weight decay with $\lambda = 10^{-4}$, applying a 0.5 dropout to the first two fully connected layers. For a discussion on weight decay regularization with *Adam* optimization see [13]. Furthermore, we randomly flip the video clips horizontally to increase the training set variability. Finally, we perform additional augmentation by extracting a random 28×28 crop from each frame, resizing it back to ×32×32.

Kernel size is set to 5×5×5 according to grid search too [4], and results of accuracy with different kernel size are given in Table 7.

In the following experiments, unless explicitly specified, the number of input frames K has been set to 32, see Table 7, for the impact of the number of input frames on the accuracy prediction.

All the source code, which is publicly available, has been written in Python, with the deep model implemented using Tensorflow 1.15 library at https://github.com/alcor-lab/MSN.

V. RESULTS

In this section, we discuss the results of the performed experiments. The results include a comparative study of available methodologies, the performance on HMDB-51 and Mini-Kinetics datasets, a comparative and ablation study on the proposed fusion methods, hyper-parameter optimization and, finally, time performances analysis.

Comparative analysis. Table 2 highlights the two classes of method, the *heavy* and *light* ones. Methods exploiting images in high resolution, obtain greater accuracy, but have to exploit computationally intensive backbone architectures. Architectures with a huge amount of parameters are also exploited by methods using *eLr* data. Table 1 associates these models with I3D-type models, showing similar computational cost and time performances.

TABLE 2. In the Table, Backbone Methods, According to State of the Art, are Compared to the Data Resolution. The Majority of Methods Exploit Either Architectures Asking for Massive Computation or Higher Resolution Input Data, Instead of MSN and a Few Other Algorithms.

Architecture	Exploited Data	Backbone
MSN	*eLR*	Three 3D conv Streams
Semi-Coupled [6]	*eLR*	Two 2D Conv Streams
ISR [18]	*LR + HR*	Custom 2D Conv
S.T.M.H.S.A. [15]	*eLR + HR*	Double I3D
TSN-BSA [14]	*eLR*	Three Stream I3D
Fully Coupled [29]	*eLR*	C3D
Multi-Siamese [17]	*eLR + HR*	Custom 2D Conv

Test on HMDB51 dataset. Our proposed MSN results are summarized in Table 3 showing the accuracy obtained on the HMDB51 dataset. These results are compared to the Semi Coupled Two-Stream Network [6] (SCTS), which uses a light-weighted two branches approach and a similar data preprocessing, and Inverse Super Resolution (ISR) [18]. MSN obtains the best performance with three streams (RGB, optical Flow and slack Mask): this demonstrates the importance of different perceptive sources to get the best results, improving the accuracy by 4.45%. Moreover, even with RGB only, MSN can achieve better results w.r.t. STCS exploiting an improved and optimized 3D architecture. We report the per-class accuracy in Figure 4, which shows excellent results (>60%) on some easy classes, like ride bike and catch, contrasting the low accuracies (<20%) on hard categories like hand-wave and kickball. In Figure 5 we report the confusion matrix.

TABLE 3. The Table Shows the Performance of MSN on the HMDB51 Dataset, Compared to the Semi Coupled Two-Stream Network and the Inverse Super Resolution One (ISR). The Reported Values are the Average Accuracies Over the Three Official Splits.

Architecture	Streams	Accuracy
Semi Coupled [6]	RGB	19.10%
Semi Coupled [6]	RGB, Optical Flow	29.20%
ISR [18]	RGB, Optical Flow	28.68%
MSN	RGB	29.66%
MSN	RGB, Optical Flow	31.11%
MSN	RGB, Optical Flow, Mask	**33.65**%

Mini-Kinetics. We tested the performance of MSN on the challenging Mini-Kinetics 200 dataset, reported in Table 4. SCTS fails to converge, reporting a very low accuracy of 5.8%: as opposed to this failure, MSN is able to effectively

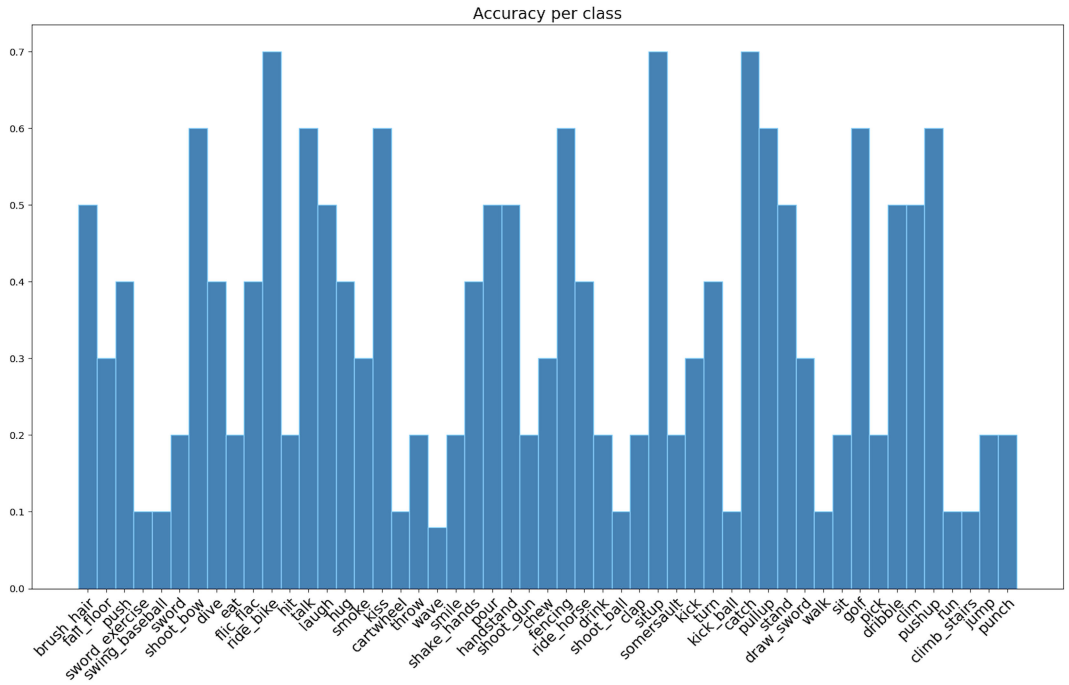


FIGURE 4. MSN per-class accuracy on the first split of HMDB51 dataset on scale 0-1.

TABLE 4. The Table Shows the Performance of MSN on the Mini-Kinetics 200 Dataset, Compared to the Semi Coupled Two-Stream Network. The Reported Values are Calculated Over the Official Validation Set.

Architecture	Streams	Accuracy
Semi Coupled [6]	RGB	5.84%
MSN	RGB, Optical Flow, Mask	**26.37%**

learn training data features, achieving an accuracy of 26.31%, despite the difficulty of a 200-categories classification task conducted on 16×12 data.

The reported results are obtained with the **Sum** fusion method, which provides the best performance, as shown in the following study.

Furthermore, we develop three sets of experiments for testing fusion methodologies, streams ablation, kernel size hyper-parameters search and time performances.

Streams ablation results are reported in Table 6. As expected, RGB stream is the most relevant source of information, providing 29.71% of accuracy. The use of optical flow or mask stream as a single input yields lower performances of $\sim(4-5)$ points. Moreover, the improvement in accuracy is steady when using two and three streams, achieving 31.11% and 34.31%, respectively.

Fusion methodologies MSN proves to be a robust network with respect to the features fusion choice, as can be seen from Table 6. Surprisingly, the fusion method **Sum**, despite its simplicity, is the approach which produces the best performance, with all the others lagging behind of about 1%.

TABLE 5. Time Performances of MSN and I3D Architectures, Respectively on a TitanX GPU and a Google Tensor Processing Unit (TPU). The Time Values Refer to the Model Inference Time Over a Single Sequence of 32 Frames.

Architecture	Streams	Device	Time
I3D	RGB	GPU	48ms
MSN	RGB	GPU	**3ms**
I3D	RGB, OF, Mask	GPU	135ms
MSN	RGB, OF, Mask	GPU	**7ms**
I3D	RGB	TPU	4100ms
MSN	RGB	TPU	**13ms**
I3D	RGB, OF, Mask	TPU	11500ms
MSN	RGB, OF, Mask	TPU	**36ms**

TABLE 6. Ablation Study of Both the Proposed RGB, Optical Flow, and Mask Streams and the Fusion Strategies Adopted for Merging Them, for the First Split of HMDB51 Dataset.

RGB	Optical Flow	Mask	Fusion Type	Accuracy
X	-	-	None	29.71%
-	X	-	None	24.43%
-	-	X	None	25.16%
X	X	-	Sum	31.11%
X	X	X	Sum	**34.31%**
X	X	X	Concatenation	33.33%
X	X	X	Interpolation	32.94%
X	X	X	SSMA	33.64%
X	X	X	Multi-Layer	33.88%

Hyper-parameters search [4] is performed on the convolutional kernel size and the length of frame sequences K, the results are shown in Table 7. From the accuracy results shown in Table 7, the $5\times5\times5$ kernel size obtains the best accuracy, with respect to other kernel sizes. Changing the

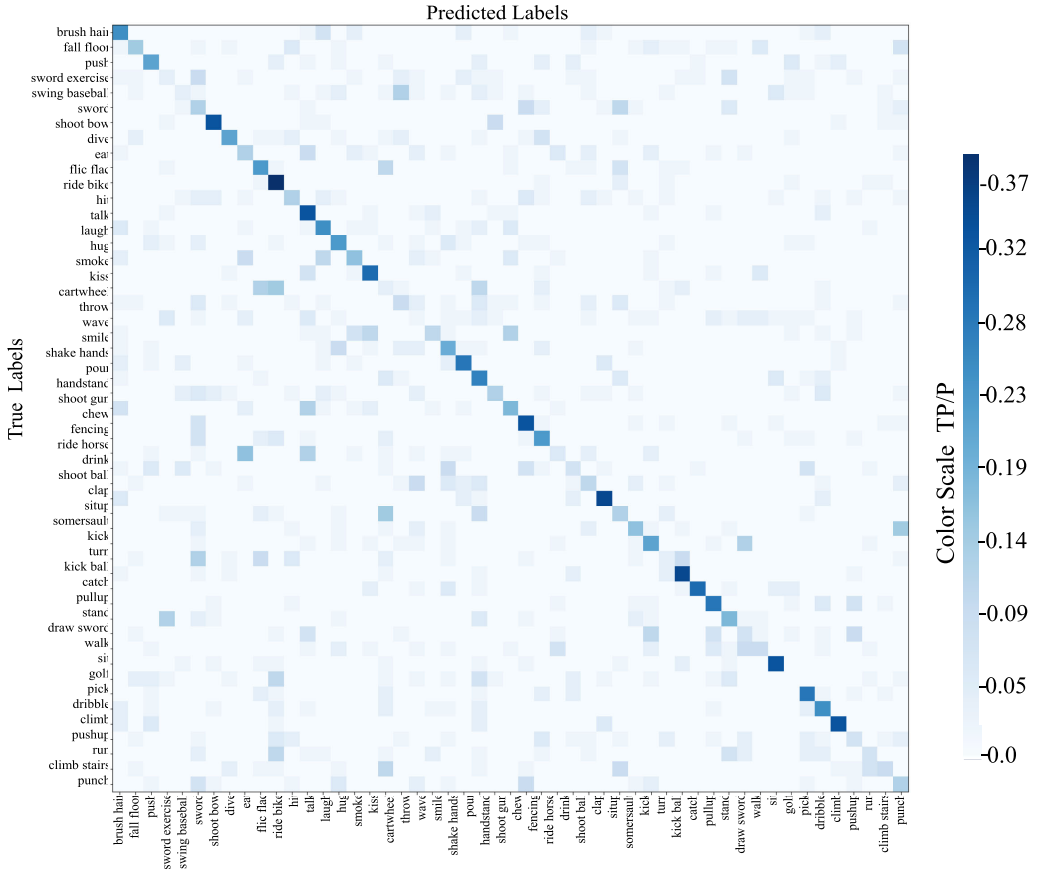


FIGURE 5. MSN multi-class confusion matrix for HMDB51 dataset. The colour scale on the right indicates the hit rate or true positive rate (TPR), approximated to two digits, over a total number of 1530 videos.

TABLE 7. Results of the Hyper-Parameters Search Performed on the Split 1 of the HMDB51 Dataset. For all Experiments, the Three-Streams Architecture Used the Sum Fusion Method: the Kernel Size Parameters Refer to Each 3D Convolutional Layer.

Kernel Size ($t \times w \times h$)	Input Frames (K)	Accuracy
$5 \times 5 \times 5$	32	**34.92%**
$8 \times 5 \times 5$	32	33.73%
$5 \times 5 \times 5$	64	34.72%
$8 \times 5 \times 5$	64	34.04%
$3 \times 3 \times 3$	64	34.29%
$8 \times 3 \times 3$	64	33.94%
$16 \times 3 \times 3$	64	33.38%

length K of the input frames barely affects the accuracy obtained by the $5\times5\times5$ kernel.

Time performances are tested on both GPU and TPU equipped devices, shown in Table 5. The three streams version of MSN can run on TPU at ∼889 *fps* as opposed to the ∼8*fps* provided by I3D. Finally, the *frames per second* of MSN when including the preprocessing time is ∼333, still faster than the real-time requirements.

VI. CONCLUSION

We propose the Multi Streams Network to boost activity recognition performances in an extremely low-resolution setting for real-time performances and sensible information protection. MSN is a deep model that effectively exploits three different perceptive sources of information, namely RGB, the optical flow and the mask streams. Performances conducted over two publicly available benchmark datasets demonstrate our approach strengths, which can obtain better accuracies w.r.t the baseline methods. Further studies will be made using pretrained networks on Kinetics, to investigate how these models exploit the three streams of information.

REFERENCES

[1] *Edge TPU Devices*. Accessed: Oct. 1, 2020. [Online]. Available: https://coral.ai

[2] S. M. Azar, M. G. Atigh, and A. Nickabadi, "A multi-stream convolutional neural network framework for group activity recognition," 2018, *arXiv:1812.10328*. [Online]. Available: http://arxiv.org/abs/1812.10328

[3] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon, "Security and privacy issues in deep learning," 2018, *arXiv:1807.11655*. [Online]. Available: http://arxiv.org/abs/1807.11655

[4] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[5] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.

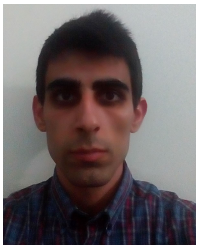
[6] J. Chen, J. Wu, J. Konrad, and P. Ishwar, "Semi-coupled two-stream fusion ConvNets for action recognition at extremely low resolutions," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 139–147.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

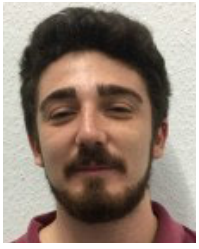
[8] T. Ha, T. K. Dang, H. Le, and T. A. Truong, "Security and privacy issues in deep learning: A brief review," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 1–15, Sep. 2020.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[10] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*. [Online]. Available: http://arxiv.org/abs/1705.06950
[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.
[13] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2017, *arXiv:1711.05101*. [Online]. Available: https://arxiv.org/abs/1711.05101
[14] D. Purwanto, R. R. A. Pramono, Y.-T. Chen, and W.-H. Fang, "Three-stream network with bidirectional self-attention for action recognition in extreme low resolution videos," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1187–1191, Aug. 2019.
[15] D. Purwanto, R. R. A. Pramono, Y.-T. Chen, and W.-H. Fang, "Extreme low resolution action recognition with spatial-temporal multi-head self-attention and knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–9.
[16] A. Roitberg, T. Pollert, M. Haurilet, M. Martin, and R. Stiefelhagen, "Analysis of deep fusion strategies for multi-modal gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 1–9.
[17] S. M. Ryoo, K. Kim, and H. J. Yang, "Extreme low resolution activity recognition with multi-siamese embedding learning," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, S. A. McIlraith and K. Q. Weinberger, Eds., New Orleans, LA, USA, Feb. 2018, pp. 7315–7322.
[18] S. M. Ryoo, B. Rothrock, C. Fleming, and H. J. Yang, "Privacy-preserving human activity recognition from extreme low resolution," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, S. P. Singh and S. Markovitch, Eds., Feb. 2017, pp. 4255–4262.
[19] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[20] S. Song, V. Chandrasekhar, B. Mandal, L. Li, J.-H. Lim, G. S. Babu, P. P. San, and N.-M. Cheung, "Multimodal multi-stream deep learning for egocentric activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 24–31.
[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[22] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
[23] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream CNN: Learning representations based on human-related regions for action recognition," *Pattern Recognit.*, vol. 79, pp. 32–43, Jul. 2018.
[24] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, vol. 128, pp. 1–47, Jul. 2019.
[25] Y. Emma Wang, G.-Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU platforms for deep learning," 2019, *arXiv:1907.10701*. [Online]. Available: http://arxiv.org/abs/1907.10701
[26] Y. Wang, Q. Wang, S. Shi, X. He, Z. Tang, K. Zhao, and X. Chu, "Benchmarking the performance and energy efficiency of AI accelerators for AI training," 2019, *arXiv:1909.06842*. [Online]. Available: http://arxiv.org/abs/1909.06842
[27] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," 2017, *arXiv:1712.04851*. [Online]. Available: http://arxiv.org/abs/1712.04851
[28] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 305–321.
[29] M. Xu, A. Sharghi, X. Chen, and D. J. Crandall, "Fully-coupled two-stream spatiotemporal networks for extremely low resolution action recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1607–1615.
[30] C. Zach, T. Pock, and H. Bischof, "A duality based approach for real-time TV-L^1 optical flow," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2007, pp. 214–223.
[31] C. Zalluhoglu and N. Ikizler-Cinbis, "Region based multi-stream convolutional neural networks for collective activity recognition," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 170–179, Apr. 2019.

PAOLO RUSSO received the B.S. degree in telecommunication engineering from the Università degli studi di Cassino, Italy, in 2008, and the M.S. degree in artificial intelligence and robotics and the Ph.D. degree in computer science from the University of Rome La Sapienza, Italy, in 2016 and 2020, respectively. From 2018 to 2019, he has been a Researcher with the Italian Institute of Techonology (IIT), Turin, Italy. He is currently an Assistant Researcher with the Alcor Laboratory, DIAG Department, University of Rome La Sapienza, Italy. His current research interests include deep learning, computer vision, generative adversarial networks, and reinforcement learning.



SALVATORE TICCA received the B.S. degree in computer science from the Università degli studi di Cagliari, Italy, in 2017. He is currently pursuing the master's degree in artificial intelligence and robotics engineering from the University of Rome La Sapienza, Italy.



EDOARDO ALATI (Member, IEEE) received the B.S. degree in computer science and control engineering and the master's degree in artificial intelligence and robotics engineering from the University of Rome La Sapienza, Italy, in 2017 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science engineering. He has been a Researcher for the Horizon 2020 European project Second Hands, University of Rome La Sapienza, since 2018. His research interests include deep learning, machine learning, and computer vision.

FIORA PIRRI (Member, IEEE) received the Ph.D. degree from Université Paris VI "Pierre et Marie Curie (UPMC). She is currently a Full Professor with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale (DIAG), University of Rome La Sapienza. She has been a principal investigators in several EU awarded projects on Cognitive Robotics and Vision. She currently leads the Alcor Laboratory. Her current research interests include cognitive robotics, vision, perception, and deep learning.