Deep Learning-Based Models for Performing Multi-Instance Multi-Label Event Classification in Gameplay Footage

Etienne Julia[a], Marcelo Zanchetta do Nascimento[b], Matheus Prado Prandini Faria[c] and Rita Maria Silva Julia[d]

Computer Science Department, Federal University of Uberlândia, Uberlândia, Minas Gerais, Brazil

Keywords: MIML Event Classification, Gameplay Footage, Super Mario, Frame-Based and Chunk-Based Data Representations, Deep Extractor Neural Networks, Fine-Tuned Backbone, Deep Classifier Neural Networks.

Abstract: In dynamic environments, like videos, one of the key pieces of information to improve the performance of autonomous agents are the events, since, in a broad manner, they represent the dynamic changes and interactions that happen in the environment. Video games stand out among the most suitable domains for investigating the effectiveness of machine learning techniques. Among the challenging activities explored in such research, it highlights that which endows the automatic game systems with the ability of identifying, in game footage, the events that other players, interacting with them, provoke in the game environment. Thus, the main contribution of this work is the implementation of deep learning models to perform MIML game event classification in gameplay footage, which are composed of: a data generator script to automatically produce multi-labeled frames from game footage (where the labels correspond to game events); a pre-processing method to make the frames generated by the script suitable to be used in the training datasets; a fine-tuned MobileNetV2 to perform feature extraction (trained from the pre-processed frames); an algorithm to produce MIML samples from the pre-processed frames (each sample corresponds to a set of frames named *chunk*); a deep neural network (NN) to perform classification of game events, which is trained from the chunks. In this investigation, *Super Mario Bros* is used as a case study.

1 INTRODUCTION

In machine learning (ML), the ability of autonomous agents to retrieve relevant information about the environment they operate in is an indispensable requisite, since it can be very useful in improving their decision-making process (Neto and Julia, 2018). One of the most relevant information to be retrieved from the environment in which agents in general operate refers to the ongoing *events*. Despite the complexity that the event definition can assume in artificial intelligence (AI) (Ravanbakhsh et al., 2017), (Yu et al., 2020), in short, events can be seen as *flags* that depict the most important things happening in the environment.

In areas like personal assistance robots, camera monitoring networks and autonomous cars, instantly being able to react to events happening in the environment is crucial. For this reason, AI research carried out in the domain of online event detection systems can be very valuable (Geest et al., 2016).

Indeed, such a challenge has motivated recent AI research on developing online event detection systems for real-world videos, be it sports, traffic, or everyday activities (Gao et al., 2017), (Geest et al., 2016) and (Xu et al., 2018).

For agents operating in dynamic environments, like videos, the events, which describe the dynamic changes and interactions that happen in the environment, represent fundamental information to guide their decision-making. In such situations, as the scenario the agents operate in can change drastically, the occurrence of the events can also be very variable.

Depending on the problem the studies carried out with the objective of retrieving events from videos cope with, they will perform a different kind of classification activity. Such variation will depend on the following factors: on the way the video scenes will be presented to the system, and on the number of events intended to be retrieved from such scenes.

[a] https://orcid.org/0000-0003-3750-4264
[b] https://orcid.org/0000-0003-3537-0178
[c] https://orcid.org/0000-0003-1468-9243
[d] https://orcid.org/0000-0001-5181-5451

Julia, E., Zanchetta do Nascimento, M., Faria, M. and Julia, R.
Deep Learning-Based Models for Performing Multi-Instance Multi-Label Event Classification in Gameplay Footage.
DOI: 10.5220/0012358900003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 3: VISAPP*, pages 110-121
ISBN: 978-989-758-679-8; ISSN: 2184-4321

In this way, the systems implemented in such studies can be divided into the following groups: 1) single instance, single label (*SISL*), where a scene is presented to the system through a single *frame* from which at most one label can be retrieved (knowing that, in this work, a *label* corresponds to a *game event*); 2) single instance, multi-labels (*SIML*), where a scene is presented through a single *frame* from which more than one label can be retrieved; 3) multi instance, single label (*MISL*), where a group of scenes is presented to the system through a set of *frames* from which at most one label can be retrieved; 4) multi instance, multi-label (*MIML*), where a group of scenes is presented through a set of *frames* from which more than one label can be retrieved.

Video games have been proving to be a very appropriate case study for investigating the effectiveness of machine learning (ML) techniques for several reasons, among which stand out: economically, they are part of the group of products that generate the highest profits in the entertainment industry (Global Data, 2021); in education, they have been showing as a friendly, efficient, and attractive tool; technically, they present a very high difficulty level, which represents a great challenge to any ML approaches used to deal with them (Faria et al., 2022).

Research aimed at producing player agents capable of recovering, from game videos, the events provoked in the game environment by the actions executed by the opponents, can be extremely useful. Indeed, for example, in the construction of player systems aimed at improving the cognitive abilities of their users, the results of the analysis of these events can be used to map the users' profile, and this profile can be used to train the player system to adjust its decision-making process so as to lead the game to situations that stimulate the development of the users' cognitive abilities.

As argued in (Faria et al., 2022), despite the fact that, frequently, the events can be directly retrieved from the game system flags (named as game logs), the following obstacles can make such information retrieval unfeasible: firstly, game engines are frequently unavailable, due to the companies' privacy concerns (Luo et al., 2019); secondly, in studies aiming at retrieving events from gameplay footage, which are not real-time games, it is not possible to count on game information provided by the game engines, even if the game platform they deal with count on open game engines.

The great applicability of systems capable of recovering events in video games, associated with this obstacle to accessing game logs, are factors that have been motivating new studies aimed at creating automatic systems with such ability.

Taking into consideration the aforementioned arguments, the authors of this paper, by way of investigation, propose two new models to perform *game event MIML* classification (that is, multi-label classification of game events) in gameplay footage of *Super Mario Bros* (Karakovskiy and Togelius, 2012). The authors' main motivation for using *Super Mario Bros* as a case study are the results presented in (Prena and Sherry, 2018), showing that such a game is quite suitable for the development of cognitive skills, especially for users with some type of weakness in the learning process, which fits well the authors' intended goals in their future works (Section 6). By way of introducing some examples of event retrieval in *Super Mario Bros*, the occurrence of the *EventJump* and *EventLand* events indicate that *Mario has just took off* and *Mario has just landed*, respectively.

In order to cope with the intended objectives, each model proposed in this work uses a chunk-based representation for the game scenes (where *chunk* refers to a set of frames) and counts on distinct and specific deep neural networks (NNs) to perform feature extraction and multi-label classification. More specifically, one of the model uses a new fine-tuned *MobileNetV2* (produced here) as a feature extractor, and the recurrent neural network (RNN), long short-term memory (LSTM), to perform the MIML classification, whereas the other one uses the same fine-tuned MobileNetV2 as backbone, and the deep MIML (Feng and Zhou, 2017) to execute the MIML classification.

Such architectures of the models were defined taking into consideration the results obtained in the studies carried out in (Faria et al., 2022) and (Feng and Zhou, 2017). More specifically, in (Faria et al., 2022) the authors investigated various architectures to perform *SISL* event classification in footage of *Super Mario* and proved that: 1) Concerning the game scenes, chunk-based representation performs better than frame-based representation; 2) Using two distinct deep neural networks to perform feature extraction and game event classification provides much better results than using just one deep neural network to execute both tasks; 3) The architecture that performed better was made up of the standard *MobileNetV2* (Sandler et al., 2018), as a feature extractor(or backbone), and the *LSTM* (Hochreiter and Schmidhuber, 1997), as a classifier network.

Concerning the second state-of-the-art work which inspired the choice of the architectures proposed in this paper, in (Feng and Zhou, 2017) the authors introduced the deep MIML neural network to perform MIML classification in the domain of images

and texts (they did not cope with footage).

Then, in short, the main contributions of this work are:

- Implementation of two new models to perform *MIML* of game event in footage of *Super Mario Bros*, thus extending the framework proposed in (Faria et al., 2022), which only coped with single event classification;
- Implementation of an automatic data generator script, which automatically generates multi-labeled frames from *Super Mario Bros* game footage (where each frame contains *zero* or more game events);
- Proposition of a pre-processing method to make the frames generated by the script suitable to be used in the training datasets;
- Production of a fine-tuned version of the standard MobileNetV2 used in the preliminary studies carried out in (Faria et al., 2022), by retraining it from frames produced by the script that were pre-processed;
- Proposition of an algorithm to produce MIML samples from the pre-processed frames (each sample corresponds to a set of frames named *chunk*);
- In order to investigate an alternative classifier deep NN to deal with *MIML* problem in gameplay footage of *Super Mario Bros*, in addition to testing the LSTM (which performed well in the single label classification scenarios investigated in (Faria et al., 2022)), the present work also tested the deep MIML neural network proposed in (Feng and Zhou, 2017) (in this later, the performance of the Deep MIML coping with footage was not investigated). Both classifier NNs were trained from the MIML samples (chunks) just mentioned.

The experiments proved the significant performance gain obtained in the models with the use of the fine-tuned MobileNetV2 instead of the standard MobileNetV2. In addition, regarding the classifiers, they showed that the LSTM model presented slightly better results than DeepMIML in terms of mean average precision and F1-Score.

2 THEORETICAL FOUNDATION

This section will present a general view of the main theoretical topics related to this work.

2.1 Multi-Instance Multi-Label Framework

The MIML framework, proposed in (Zhou et al., 2012), is defined by each data example being comprised of multiple instances, associated with multiple labels. Traditionally in ML, each instance of data is associated with a single label, or even, each instance with multiple labels. However, for problems involving complicated examples with multiple semantic meanings, using more than one instance to represent them can allow for additional inherent patterns to the data to become more clear. Then, in these situations, the MIML framework can be a more natural and appropriate way to represent the data.

An important idea in multi-instance learning are sub-concepts. For example, when considering a broad concept like "*Africa*", it can be hardly described by just one aspect, which requires a group of cultural and environmental identifiers that make possible such description. In this case, the broad concept "*Africa*" could be identified, for example, from images that combine low-level concepts such as a grassland environment, lions and trees. It is important to note that these low-level concepts, called *sub-concepts*, alone are not necessarily sufficient to describe the broad concept "*Africa*", however, when combined, they become capable of doing that.

2.1.1 Multi-Label Evaluation Metrics

In traditional supervised learning, as mentioned in the previous subsection, a single instance is associated with a single label. This allows for an *accuracy* metric (the percentage of examples correctly classified) to often be a good enough indicator of performance (Zhou et al., 2012). However, in multi-label situations, the goal is not to identify a single label, but rather to correctly classify the highest amount possible from a group of labels. Then, for example, it is better for a model to identify 4 out of 5 labels correctly, than getting 2 out of the same 5 labels. This has to be taken into account by a multi-label evaluation metric for it to be an effective performance indicator.

The evaluation metrics that were relevant to this work are explained below:

- Hamming loss: a loss metric that represents the fraction of labels that are incorrectly predicted, be it a correct label that is missed, or a wrong label that is predicted. Considering the function $hamming_loss(h) = 0$, the lower the value of $hamming_loss(h)$ the better the performance of h.
- Mean average precision: the average precision (AP) is a relation between the class precision

and class recall. Precision (P) indicates how many predictions were correct and recall (R) indicates how many of all instances of a class were predicted by a model. The average precision corresponds to the area under the graph $precisionX recall$. Since the average precision corresponds to a single class, the mean average precision (mAP) corresponds to the average APs over all classes.

- F1 score: the F1 score, also is a relation between class precision and class recall. However, it corresponds to the harmonic mean between the two, so: $F1 = 2 * \frac{P*R}{P+R}$. Like mAP, the F1 score also references a single class, so in a multi-label context, this metric corresponds to the average F1 score of all classes.

2.2 Deep MIML Network

As previously mentioned in section 2.1, a lot of real-world applications greatly benefit from being represented in a MIML context. The *deep MIML* network (Feng and Zhou, 2017) is a model that adds an automatic instance generator and a sub-concept learning structure to the traditional neural network formation. The structure of the deep MIML network (Figure 1) consists on a feature-based-instance generator module (usually a CNN network), followed by a *sub-concept layer*, which is essentially a classifier that matches the scores between instances and sub-concepts for every label. This approach allows for an instance-label relation discovery that works very well in a MIML framework. Then, the role of the *sub-concept layer* is to automatically abstract a set of sub-concepts (Subsection 2.1) present in the received instances (images) that will be useful in helping to identify the labels that occur in these instances. It means that the deep MIML, in addition to performing an automatic extraction of features, also performs an automatic extraction of sub-concepts that will help in the classification task.

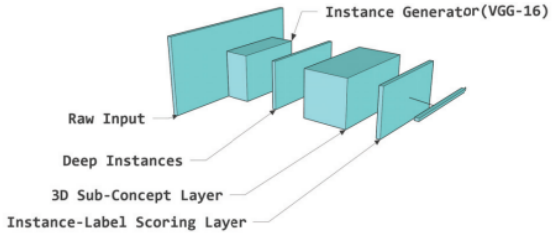

Figure 1: Representation of a deep MIML network, image from (Feng and Zhou, 2017).

2.3 Super Mario Bros

The game *Super Mario Bros* was first released in 1985 by the company Nintendo. In it, the player plays as a plumber called Mario and has to traverse a series of stages to reach and save a princess. Even though it is a very simple premise, each stage presents a different set of obstacles and enemies that require precise actions from the player in order to advance. These actions are: walking, running, jumping and throwing a fireball. All of these actions trigger different *game events*, that represent the interactions the player has with the environment through his actions. Noteworthy here is that the same action can trigger more than one game event in the game scenes. The main game events related to *Super Mario Bros* are listed in section 4.1.

From a visual standpoint, the game has very simple graphics in a ”pixelated” style. The game also presents a 2D (two-dimensional) point of view, which lacks the object depth that is present in 3D games (Roettl and Terlutter, 2018) as can be seen in Figure 2.

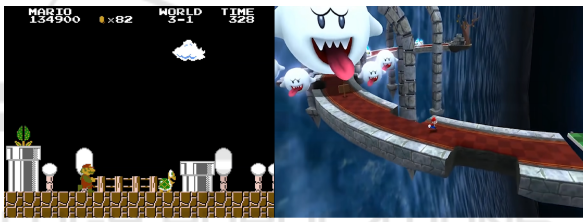

Figure 2: *Super Mario Bros* on left (2D), Super Mario Galaxy on right (3D).

3 RELATED WORKS

This section is structured as follows: Subsection 3.1 presents in more detail both state-of-the-art works whose results inspired the definition of the architectures of the models proposed in this paper, whereas Subsection 3.2 resumes other state-of-the-art works related to SISL or MIML classification (in games or other domains involving images).

3.1 Basic Related Works

This subsection summarizes the state-of-the-art works whose architectures serve as a basis for the models proposed here. The work (Faria et al., 2022) is a precursor of the present proposal in which the authors investigated 26 DL-based models to identify events in gameplay footage depicting Super Mario runs (Mario AI Framework (Karakovskiy and Togelius, 2012)). Their main objective was to find the

most adequate architecture and game scene representation to perform *SISL* classification of game events in footage of Super Mario. One of the investigated models was similar to the single standard AlexNet-based model proposed in (Luo et al., 2018), where the AlexNet architecture performed both the feature extraction and the classification. The remaining models combined a CNN automatic feature extractor (MobileNetV2, ResNet50V2, VGG16 or AlexNet) - to produce a concise and adequate feature-based representation for the game scenes - and a NN game event classifier (long short-term memory (LSTM), gated recurrent unit (GRU) or fully-connected layers (FCL) - to identify the game event occurring in the feature-based representation produced by the CNN extractor. Concerning the game scene representation at the input of the CNN extractor, two kinds of data were investigated: individual frame and chunk (a set of consecutive frames). The major contribution provided by (Faria et al., 2022) consists on proving that, concerning the problem of event classification in gameplay footage: firstly, the chunk-based representation performs better than the frame-based representation; secondly, the models which use distinct deep neural networks to perform feature extraction and event classification provide much better results than those in which a single CNN carries out both activities; finally, the model that presented the best performance combines the *standard MobileNetV2* (Sandler et al., 2018), as backbone, and the *LSTM* (Hochreiter and Schmidhuber, 1997), as classifier network (this better model uses a *chunk*-based representation for the game scenes), which motivated the authors of this paper to propose a new model that extends and improves the best model produced in (Faria et al., 2022) as follows: proposing a new *MobileNetV2+LSTM* - based model able to face the more complex problem of performing MIML classification of events in gameplay footage; implementing an automatic data generator script to produce the frames from which the training dataset will be constructed; proposing a method to build multi-labeled datasets to train the new models; finally, by replacing the standard MobileNetV2 used in the best model of (Faria et al., 2022) with a fine-tuned version of MobileNetV2, which is produced by training the standard CNN from the frames that are generated by the automatic data generator script.

The other state-of-the-art work closely related to the architectures proposed in this paper is (Feng and Zhou, 2017), where the authors, based on the Multi-Instance Multi-Label framework proposed in (Zhou et al., 2012) (which is widely used to structure classification problems across many research fields), created the *deep MIML network*. In (Feng and Zhou, 2017), the deep MIML (resumed in Section 2.2) was trained from numerous and varied real datasets, so as to be able to perform MIML classification of events occurring in images and texts (in such work, the authors did not cope with footage). In this way, the second model proposed in this paper has as purpose to investigate the performance of the Deep MIML dealing with MIML classification of events occurring in game footage. For this, here the authors replace the LSTM of the first proposed model with the deep MIML.

3.2 Other Related Works

This subsection resumes other related works involving SISL or MIML classification in games or domains involving images. In (Song et al., 2018), the authors explored the usage of CNNs on multi-label image classification problems. They proposed a deep multi-modal CNN, that was designed for MIML learning, and whose main appeal was combining the benefits of the MIML framework with the image classification capabilities of CNNs in order to solve the problem of weakly labeled images in classical datasets, like ImageNet, in which images presenting multiple object classes are labeled with just a single label. Differently from the approaches implemented in this paper, in (Song et al., 2018): the authors uses just a single deep neural network to perform both the feature extraction and the MIML classification; and the authors do not face the challenge of performing MIML classification in footage. The work (Luo et al., 2018) proposed a DL based approach to deal with SISL game event classification in footage of the *Gwario* (a Super Mario's clone). Also differing from this work, in (Luo et al., 2018): a single CNN (AlexNet) was used for both feature extraction and classification; the authors only investigated the frame-based representations for the game scenes; and, finally, the models were trained from a manually labeled Gwario game dataset.

4 ARCHITECTURE OF THE MODELS

This section details the models implemented in this work to tackle *MIML* event classification in gameplay footage of *Super Mario Bros*. These proposed approaches are based on the association of a backbone CNN and a classifying deep NN, where the former must automatically extract the most relevant features corresponding to the game scenes (represented through chunks), and the later has as its purpose to identify the game events occurring in such scenes, as shown in Figure 3.

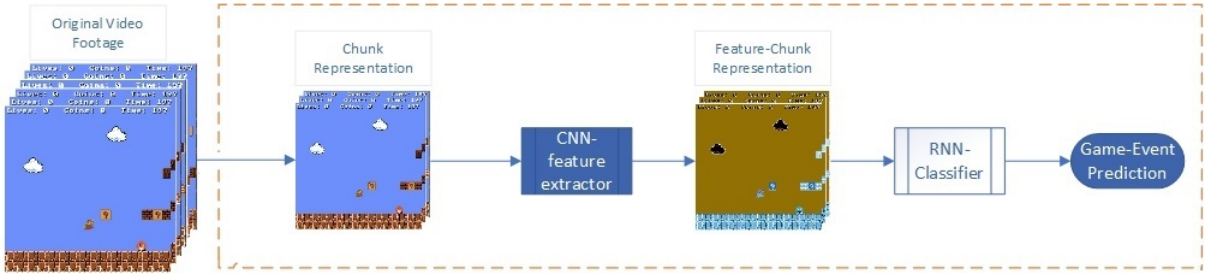

Figure 3: General architecture.

The models were implemented in the PyTorch API.

The approaches were created so as to extend the results obtained in the following state-of-the-art works: 1) In (Faria et al., 2022) (Section 3.1), the authors investigated various models to perform SISL game event detection in footage of *Super Mario Bros* (differently from the present paper, such work did not cope with MIML classification). These investigated models differed from each other according to the following aspects: first, in the way of representing the game scenes: frames or chunks; secondly, in the composition of the architecture of the models: one of the architectures was composed of a single deep network to perform both feature extraction and single event classification and a second alternative architecture composed of two distinct networks, where a CNN performed feature extraction and another network (a deep RNN or a multi-layer NN) performed the single label classification. The winner model was made of a MobileNetV2 feature extractor combined with a classifier LSTM, and used chunks to represent the game scenes; 2) In (Feng and Zhou, 2017), the authors proposed the DeepMIML 2D sub-concept layer to perform multi-label classification in images and texts (distinctly from this work, they did not investigate the performance of DeepMIML executing *MIML* classification in videos).

Then, in order to cope with the objective of performing *MIML* classification in game footage of *Super Mario Bros* and taking into consideration the results obtained in such state-of-the-art-works, both models proposed in this work, additionally to use chunks to represent the game scenes, count on two distinct NN to perform feature extraction and MIML classification. The architecture of these models can be resumed as follows:

- Both models count on an automatic data generator (script): the script has as its purpose to generate the labeled data (frames) that will be used to build the training datasets. For this, the script uses the Mario AI Framework (Karakovskiy and Togelius, 2012) to produce game footage from real games. Then, from such footage, the script generates the following elements: the frames representing the game scenes and a set of .cvs files storing the labels corresponding to these frames.
- CNN backbone: in both models, a new *fine-tuned MobileNetV2* was implemented with the purpose of replacing and enhancing the *standard MobileNetV2* backbone used in (Faria et al., 2022). Such a new version was produced by retraining the *standard MobileNetV2* from a dataset composed of the frames produced by the automatic data generator (script);
- Multi-label classifiers: by way of investigation, one of the models uses the LSTM, which proved to be the best single label classifier in (Faria et al., 2022), whereas the other one uses the Deep MIML proposed in (Feng and Zhou, 2017), in such a way as to evaluate to which extent it performs well in making *MIML* in videos.

The details of such *MIML* classification from gameplay footage will be explained in detail in the following sub-sections.

4.1 A Script to Automatically Generate Labeled Frames from Game Footage

As this work copes with MIML classification, it has to count on a significant quantity of multi-labeled samples to train the deep classifier NNs (Section 4.3.2). Then, it was necessary to implement a script that automatically retrieved data from gameplay footage. For this, the script, through adequate resources provided by the Mario AI Framework (Karakovskiy and Togelius, 2012), produced the gameplay footage from which the following elements were extracted: the data corresponding to frames that represent the game scenes, and some .cvs files containing the labels (game events) associated to these frames. It is interesting to remember (Section 2.3) that the game events appear as a consequence of the user actions and that

the same action can generate more than one game event.

The footage is composed of 75 videos, each one containing the recording of the same group of 15 levels of *Super Mario Bros* played by a different competitor picked up from a set of 5 players, among which 4 are humans and one is the automatic player available in the Mario AI Framework.

The script also had to define both the *frame rate* of the footage - which represents how many times the game scene refreshes per second of the game - and the set of events that is adequate to label the data samples (frames).

Concerning the *frame rate*, the footage was captured at 30 frames per second. It means that the time interval between two consecutive frames is about just 0.03 seconds, which very frequently makes the script generate frames devoid of events (in fact, as the time interval between 2 consecutive frames f_1 and f_2 is very small, usually there is no time for a user action - which would introduce a new event in f_2 - to be executed between both frames). It is interesting to note that, in this case, reducing the frame rate in order to reduce the number of frames with no event is not a good alternative, since it would cause a problem even more serious: a too low capture of information related to the game scenes, which would compromise the training of the models.

Then, the frames generated by the script can present zero or more events.

In order to label the frames, the script used the same 12 game events pointed out by the framework itself as being fundamental to model the dynamics of a match (such events stand out among the 270 variables generated by the Mario AI Framework to describe a given data scene, which include game events, coordinates, and others) (Faria et al., 2022).

The game situations corresponding to these 12 game events are:

- EventBump: Mario bumps his head on a block after jumping;
- EventCollect: Mario collects a coin;
- EventFallKill: an enemy dies by falling out of the scene;
- EventFireKill: Mario kills an enemy by shooting fire at it;
- EventHurt: Mario takes damage after being hit by an enemy;
- EventJump: Mario performs a jump;
- EventKick: Mario kicks a Koopa shell;
- EventLand: Mario lands on the ground after having jumped;
- EventLose: Mario loses the game level, which can be caused by the player dying or the time running out;
- EventShellKill: Mario kills an enemy by kicking a Koopa shell at it;
- EventStompKill: Mario kills an enemy by jumping on top of it;
- EventWin: The player completes the current level (illustrated as an example in Figure 4).

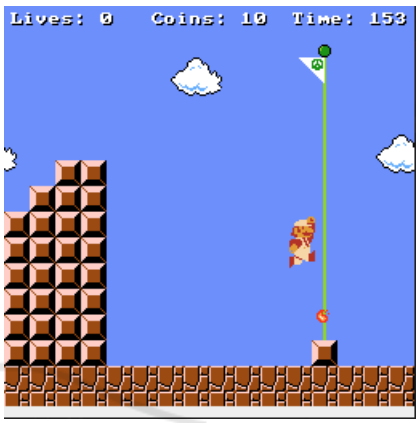

Figure 4: Example of an *EventWin* frame.

4.2 Data Pre-Processing

The models were trained from datasets built from pre-processed frames that were generated by the automatic script. The purpose of the pre-processing is to refine the row samples that were produced by the script, making them more suitable to enhance the quality of the training of the proposed models. In short, the following pre-processing stages were implemented: 1) Definition of an adequate capture window: normally, whenever the user executes an action, the Super Mario AI Framework registers all the events associated with that action in the frame that depicts the game scenario at the exact moment in which such an action was executed. However, for some kinds of events, such an annotation is not adequate, since, visually, the effect of such execution (that is, the visualization of the game events) will only be noticed in a later frame. Due to this, the first pre-processing stage has as its purpose to define an adequate custom capture window for each kind of event label. For example, specifically for the event *EventBump*, instead of annotating its occurrence in the frame f where it was fired off (as the framework does), here, through this pre-processing stage, it will be registered in the frame that succeeds f; 2) Frame resizing: with the purpose of reducing the feature extraction runtime, in this second pre-processing step, each frame was resized from

its original size to 224×224 pixels, and a pixel-wise normalization was applied. All three RGB color channels were kept.

4.3 Training Datasets

In order to train the models proposed in this work, two datasets had to be created: a frame-based dataset and a chunk-based dataset.

For this, firstly, once the process of generating training samples has been completed by the script (Subsection 4.1), all labeled row frames with *at least one* event that were generated were pre-processed (Section 4.2) and stored in a primary dataset. Next, as such a dataset was originally very unbalanced due to the fact that some events naturally occur much more than others in the footage from which the data were retrieved, it had to be submitted to an adequate balancing strategy. As, due to intrinsic characteristics of the *Super Mario Bros* game, about 98% of the frames with *at least one event* are single-labeled instances (that is, they present just *one* event), it was possible to use a simple oversampling balancing in which the frames belonging to the underclasses were randomly duplicated until a more even distribution was reached. At the end, this pre-processed and balanced primary dataset was composed of 10,000 examples, among which only 218 were multi-labeled frames (that is, frames with more than one event). As shown in the following subsections, this pre-processed and balanced primary dataset represents the basis for creating the training databases.

4.3.1 Frame Dataset

In order to obtain a more effective feature extractor method, here the standard MobileNetV2 pre-trained from ImageNet (Sandler et al., 2018) had to be retrained. To speed up such retraining, it was performed from a *Frame Dataset* exclusively made up of the 9,782 single labeled frames present in the pre-processed and balanced primary dataset aforementioned. The idea, in this case, is to improve the backbone NN by retraining it from images specifically corresponding to the problem it will cope with, that is, the *Super Mario Bros* images.

4.3.2 A Special Algorithm to Create the Multi-Labeled Samples of the Chunk Dataset

In order to build the *Chunk Dataset* to train the MIML classifiers of game events of both models, it was necessary to create an algorithm to produce a significant quantity of multi-labeled samples, since, due to special characteristics of *Super Mario Bros* game, a very large part of the frames generated by the script are devoid of events (Section 4.1) or, then, present just a single event (as discussed in the beginning of Section 4.3).

Each multi-labeled sample (*or chunk*) of the *Chunk Dataset* produced by such an algorithm corresponds to a sequence of *seven* pre-processed frames, where each one contains *zero* or more game events, as detailed in the following.

Each chunk C_f of the *Chunk Dataset* was generated through the application of the following algorithm to one of the 10,000 frames f which makes up the pre-processed and balanced primary dataset (it means that 10,000 chunks were generated since all the frames of the primary dataset have at least one game event (Section 4.3)):

1. Take a frame f from the pre-processed and balanced primary dataset;
2. Find f in the sequence of row frames produced by the generator script; in the same sequence, take the three frames that precede f and the three frames that follow it (it is interesting to note that such frames can be devoid of event);
3. Apply the same pre-processing described in 4.2 to all the frames taken in the previous step that present no event (which had not yet been pre-processed, since they do not belong to the primary pre-processed and balanced dataset). Note that all the remaining row frames taken in the previous step (that is, those with at least one event) have already been pre-processed since they belong to the primary pre-processed and balanced dataset;
4. Build the chunk C_f by grouping, in the same order they appear in the sequence produced by the generator script, the three frames that precede f, f itself, and the three frames that follow f (and all these 7 frames must already be in the pre-processed form). It is interesting to point out that such a strategy of generating chunk-based data by grouping consecutive frames retrieved from the footage by the script is very useful for two reasons: first, this produces a kind of synthesis of a temporal fraction of the game scenes, which will contribute to improving the perception of the game by the deep classifier NN and, consequently, their performance; secondly, it allows for significantly increase the number of multi-labeled instances in the Chunk Dataset (as detailed in the following), which is very useful to enhance the multi-label classifier training.

5. Label the chunk C_f with the set of labels occurring in the 7 frames that compose it.

This algorithm proposed here to generate multi-labeled samples proved to be quite effective, since, despite the fact that a very large part of the frames with which it operated were devoid of events or, then, had just a single event, the algorithm managed to produce a very expressive quantity of multi-labeled samples to train the classifiers. In fact, among the $10,000$ chunks that make up the *Chunk Dataset*, $3,791$ correspond to multi-labeled examples (with two or more labels), thus enabling the training of the MIML classifiers (which would have been impossible to do with the pre-processed and balanced primary dataset, which just counts on 218 multi-labeled samples, as explained at the beginning of section 4.3).

Table 1 summarizes information regarding frame and chunk Datasets.

Table 1: Summarization of the Training Datasets.

	Total Instances	Single Label Instances	Multi Label Instances
Frame-Dataset	9782	9782	0
Chunk Dataset	10000	6209	3791

4.4 Construction of the Fine-Tuned Backbone

Before presenting how the fine-tuned backbone NN used here was produced, it is important to remember that the architecture of both models proposed in this paper to perform MIML classification of game events in footage of *Super Mario Bros* is inspired by the best model obtained in the preliminary studies carried out in (Faria et al., 2022), in which the feature extraction and the classification processes are performed by distinct deep NNs, and the standard MobileNetV2 proved its effectiveness in performing feature extraction in SISL classification of the game event in footage of *Super Mario Bros*. For this, the training of the feature extractor and the classifier NNs had to be executed separately, as explained in the following.

The fine-tuned MobileNetV2 backbone proposed here was created in two steps: 1) Retraining, on the *Frame Dataset* presented in Section 4.3.1, of the standard MobileNetV2 (Sandler et al., 2018). The motivation, in this case, is the fact that the ImageNet dataset, from which the standard MobileNetV2 was trained, is composed of real-world images that differ a lot from the visual aspects of Super Mario's game scenes. Thus, the retraining of MobileNetV2 from the *Frame Dataset*, whose instances correspond to frames reporting the real visual aspects of the game scenes, will allow for producing a more specialized backbone to deal with *Super Mario Bros*. Such retraining was performed with a 5-fold cross-validation method for up to 100 epochs, or until there was no improvement on the loss for five consecutive epochs. The optimizer used was Adam (Kingma and Ba, 2015), the loss parameter was categorical cross-entropy, and the learning rate was equal to 0.0005, following what was used in (Faria et al., 2022); 2) Shortening of the architecture of the specialized backbone obtained in *step 1* by cutting off its fully connected layer, responsible for classification (in such a way as to keep only the extractor layers).

This way, such a shortened architecture, - from this point on, referred to just as *fine-tunned MobileNetV2* -, will be able to produce, at its extractor output layer, the feature-based representation of each chunk that must be presented at the input layer of the deep classifier NN, with the purpose of training it, as presented in the next section.

Finally, it is interesting to point out that this fine-tuned MobileNetV2 model produced here has the same number of features (at the output extract layer) and parameters as the extractor part of the pre-trained MobileNetV2 (Sandler et al., 2018), that is, 1280 features and 2,257,984 parameters.

4.5 Classifiers

In order to find an adequate algorithm to perform classification in game-play footage, this work investigated two distinct deep NN classifiers: the sub-concept layer of the DeepMIML (Feng and Zhou, 2017) and the LSTM (Hochreiter and Schmidhuber, 1997).

The motivation here to investigate the sub-concept layer of the DeepMIMIL as a classifier NN in one of the proposed models is the following: as resumed in Subsection 2.2, it corresponds to the classifier portion of the DeepMIML NN architecture, which was conceived to operate with MIML problems (Feng and Zhou, 2017). The effectiveness of such a classifier layer to automatically generate the sub-concepts that will help to identify each label makes it quite suitable for dealing with MIML problems. Further, the fact that the present work uses multiple frames with multiple labels (chunks) as the basis for representing the game scenes at the input of the NN classifiers, allows for including the problem it deals with in the class of the MIML problems (Zhou et al., 2012), which naturally justifies the proposition of the hypothesis that the DeepMIML can be an interesting approach to be included among the classifier deep NNs to be investi-

gated herein.

With respect to the LSTM model, it was selected for the following reasons: firstly, because it proved to be very effective as an NN classifier in the winner model obtained in the preliminary studies related to single event classification in *Super Mario Bros* gameplay footage carried out in (Faria et al., 2022). Such a good result is due to the fact that, in LSTM, the intermediary neurons are replaced with memory cells that allow storing the persistence of relevant information in the NN across the processing of different instances. The second reason that motivated the authors of this paper to investigate the LSTM in one of their models is that in the very work that proposed the Deep MIML (Feng and Zhou, 2017), in order to evaluate it, the authors carried out experiments in which they compared its performance in executing MIML classification in images with that of LSTM (differently from this work, they did not investigate the performance of LSTM and Deep MIML operating as *MIML* classifiers in footage).

Both classifiers were trained in the following way: firstly, each training chunk was retrieved from the chunk dataset (Section 4.3.2) and presented at the input layer of the fine-tuned MobileNetV2; next, the feature-based representation generated, for each chunk, at the output layer of such backbone, was presented at the input layer of the NN classifier in order to be classified.

This way, in both models, the fine-tuned MobileNetV2 model provides the feature-based representation for every chunk from the *Chunk Dataset* that will be used to train the classifiers, as mentioned in Section 4.4. More specifically, concerning the deep MIML-based model, it is interesting to point out that the fine-tuned MobileNetV2 model plays the role of the feature-based-instance generator module mentioned in Section 2.2.

Further, the output layers of both classifiers represent the 12 game events (labels) mentioned in Section 4.1.

Both classifiers were also implemented according to a 5-fold cross-validation method, with a 0.0005 learning rate used for training and testing. The chosen optimizer was *dadelta* (Zeiler, 2012).

5 EXPERIMENTS AND RESULTS

This section presents the experiments carried out to evaluate the performance of both models proposed in this paper. Such an evaluation will be done through the following two experiments: the first one aims to compare the efficacy of the fine-tuned backbone produced here and the standard MobileNetV2. The second evaluative experiment has as its purpose to compare the performance of both MIML classifiers of game events investigated in this paper, that is, LSTM and DeepMIML.

5.1 Evaluation of the Feature Extractor

This subsection has as its purpose to make a comparative evaluation between the performance of the fine-tuned MobileNetV2 model produced here (Section 4.4) and the standard MobileNetV2 model.

Such a comparison was made through the same evaluative parameters used in the state-of-the-art (Faria et al., 2022) to compare the performance among backbone NNs, that is, the *Mean Accuracy* and the *Mean Loss*. Table 2 shows that the fine-tuned MobileNetV2 model proposed in this work (accuracy 90.81% and Mean Loss 0.283%) performs better than the standard MobileNetV2 model (accuracy 74.69% and mean loss 0.702%). This makes sense, considering that ImageNet, from which the standard MobileNetV2 model was trained, is a dataset comprised of real-world images, whereas the *Frame Dataset* used to train the fine-tuned MobileNetV2 model was composed of images related to the problem faced here.

These good results obtained in this first experiment confirm the following contributions of this work: 1) The data generator script created and the preprocessing method proposed were successful, since the labeled frames produced by both, in fact, allowed for building an appropriate Frame Dataset to train the fine-tuned MobileNetV2 model; 2) The *fine-tunned MobileNetV2* model, in fact, improves the performance of the standard MobileNetV2, which made it be chosen to perform feature extraction in both models proposed.

Table 2: Mean Accuracy and Loss Results

	Mean Accuracy	Mean Loss
MobileNetV2	74.69%	0.702
MobileNetV2-FineTuned	90.81%	0.283

5.2 Evaluation of the Classifiers

As explained before, this work proposes two distinct models to perform MIML event classification in-game footage: 1) a first one, combining the fine-tuned MobileNetV2 model as a feature extractor and the sub-concept layer of the DeepMIML as a classifier, named *fine-tuned-MobileNetV2 + DeepMIML-SubConcept-Layer*; and 2) a second model, combin-

ing the fine-tuned MobileNetV2 model as a feature extractor and LSTM as a classifier, named *fine-tuned-MobileNetV2 + LSTM*.

This section presents the results obtained in the second experiment conceived to make a comparison between the performance of the LSTM and the Deep MIML-based models.

This experiment was carried out based on the same parameters that have been used in the state-of-the-art (Feng and Zhou, 2017) to compare the performance of *MIML* classifiers, that is: the *Mean Average Precision*, the *F1-Score* and the *Hamming Loss* (Section 2.1.1).

The results obtained in the experiments are presented in Table 3.

Table 3: Classifier Results

	mAP	HammingLoss	F1
DeepMIML Sub-Concept Layer	87.92%	0.014	0.91
LSTM	89.33%	0.011	0.93

The results show that the LSTM classifier performs a little better than MIML in all evaluated metrics. In fact, the *Mean Average Precision*, the *Hamming Loss* and the *F1-Score* produced by the former and the latter were, respectively: 89.33%, 0.011%, 0.93, and 87.92%, 0.014%, 0.91. It is interesting to note that the same slight performance superiority of the LSTM model over the DeepMIML model observed here, dealing with *MIML* classification in-game footage, was also observed in (Feng and Zhou, 2017), where the authors compared the performance of models based on LSTM and on MIML classifiers coping with problems related to making *MIML* classification in images.

The satisfactory results produced by both models in this second experiment, in addition to corroborating the gains obtained with the data generator script and the pre-processing method proposed here (since, as shown in section 4.3.2, the Chunk Dataset used to train the classifier NNs was also produced from them), also confirm the following contributions: 1) The efficiency of the algorithm that produce MIML samples (named as *chunks*) from the pre-processed frames (since the Chunk Dataset is made up of these chunks); 2) The effectiveness of both deep NNs trained to perform MIML classification of game events in footage: the LSTM and the deep MIML.

6 CONCLUSIONS AND FUTURE WORKS

In order to extend the current state of the art in the domain of performing *MIML* classification of game events in gameplay footage (using *Super Mario Bros* game as a case study), this work proposed two distinct deep learning models, in which the instances are represented by chunks and the feature extraction is made by a fine-tuned MobileNetV2 model produced in this study. With respect to the deep NN classifier, the first model uses the sub-concept layer of the DeepMIML - so far, only tested in the domain of images and texts (Feng and Zhou, 2017) -, whereas the second one uses the LSTM model - only investigated in (Faria et al., 2022) as a *MISL* approach to perform single label classification in footage of Super Mario. In order to produce such models, in this work the following elements had to be implemented: 1) A data generator script to automatically produce multi-labeled frames from game footage (where the labels correspond to game events); 2) A pre-processing method to make the frames generated by the script suitable to be used in the training datasets; 3) An algorithm to produce MIML samples from the pre-processed frames (each sample corresponds to a set of frames, named *chunk*), and will be used to train the deep classifier NN). The experiments confirmed the higher performance reached by the fine-tuned MobileNetV2 compared to the standard MobileNetV2 and also proved that both models satisfactorily succeed in the task of performing MIML classification of game events in footage, even though the *fine-tunned-MobileNetV2 + LSTM* model performed a little better than the *Fine-tunned-MobileNetV2 + DeepMIML-SubConcept-Layer* model.

Taking into consideration the fact that video games have also been extremely used in the fields of medicine and psychology as valuable tools to study behavior, improve motor skills (especially hand-to-eye coordination), and assist with the treatment of developmental disorders (Squire, 2003), (Janarthanan, 2012), (Kozlov and Johansen, 2010), (Boyle et al., 2011), as future work, the authors intend to use the models produced herein in the implementation of *Super Mario Bros* player agent endowed with the ability to stimulate the cognitive and/or motor skills of patients with some kind of syndrome (such as Down syndrome), in the following way: the events identified in the gameplay footage will represent the basis to perceive the actions performed by the game users and, through them, to map their possible cognitive weaknesses. From this information, the agent must adapt its decision-making engine to lead the game into situ-

ations that stimulate the cognitive development of its users. The authors also aim to explore deep learning models based on transformers (Khan et al., 2022) to extract high-level representations from gameplay footage.

AKNOWLEDGMENTS

To CAPES, for financial support.

REFERENCES

Boyle, E., Connolly, T. M., and Hainey, T. (2011). The role of psychology in understanding the impact of computer games. *Entertainment Computing*, 2(2):69–74. Serious Games Development and Applications.

Faria, M. P. P., Julia, E. S., Nascimento, M. Z. d., and Julia, R. M. S. (2022). Investigating the performance of various deep neural networks-based approaches designed to identify game events in gameplay footage. volume 5, New York, NY, USA. Association for Computing Machinery.

Feng, J. and Zhou, Z.-H. (2017). Deep miml network. In *AAAI*.

Gao, J., Yang, Z., and Nevatia, R. (2017). RED: reinforced encoder-decoder networks for action anticipation. *CoRR*, abs/1707.04818.

Geest, R. D., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., and Tuytelaars, T. (2016). Online action detection. volume abs/1604.06506.

Global Data (2021). Video games market set to become a 300bn-plus industry by 2025. https://www.globaldata.com/video-games-market-set-to-become-a-300bn-plus-industry-by-2025.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Janarthanan, V. (2012). Serious video games: Games for education and health. In *2012 Ninth International Conference on Information Technology - New Generations*.

Karakovskiy, S. and Togelius, J. (2012). The mario ai benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):55–67.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kozlov, M. D. and Johansen, M. K. (2010). Real behavior in virtual environments: Psychology experiments in a simple virtual-reality paradigm using video games. *Cyberpsychology, Behavior, and Social Networking*.

Luo, Z., Guzdial, M., Liao, N., and Riedl, M. (2018). Player experience extraction from gameplay video. *CoRR*, abs/1809.06201.

Luo, Z., Guzdial, M., and Riedl, M. (2019). Making cnns for video parsing accessible. *CoRR*, abs/1906.11877.

Neto, H. C. and Julia, R. M. S. (2018). Ace-rl-checkers: decision-making adaptability through integration of automatic case elicitation, reinforcement learning, and sequential pattern mining. *Knowledge and Information Systems*, 57(3):603–634.

Prena, K. and Sherry, J. (2018). Parental perspectives on video game genre preferences and motivations of children with down syndrome. *Journal of Enabling Technologies*, 12:00–00.

Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., and Sebe, N. (2017). Abnormal event detection in videos using generative adversarial nets.

Roettl, J. and Terlutter, R. (2018). The same video game in 2d, 3d or virtual reality – how does technology impact game evaluation and brand placements? *PLOS ONE*, 13:1–24.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Song, L., Liu, J., Qian, B., Sun, M., Yang, K., Sun, M., and Abbas, S. (2018). A deep multi-modal cnn for multi-instance multi-label image classification. *IEEE Transactions on Image Processing*, 27(12):6025–6038.

Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming*, 2:49–62.

Xu, M., Gao, M., Chen, Y., Davis, L. S., and Crandall, D. J. (2018). Temporal recurrent networks for online action detection.

Yu, M., Bambacus, M., Cervone, G., Clarke, K., Duffy, D., Huang, Q., Li, J., Li, W., Li, Z., Liu, Q., Resch, B., Yang, J., and Yang, C. (2020). Spatiotemporal event detection: a review. *International Journal of Digital Earth*, 13(12):1339–1365.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhou, Z.-H., Zhang, M.-L., Huang, S.-J., and Li, Y.-F. (2012). Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320.