WE DIDN'T START THE FIRE

Ivanina Ivanova, Abhay Mathur
Institut Polytechnique de Paris
{abhay.mathur, ivanina.ivanova}@ip-paris.fr

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

***K*eywords** First keyword · Second keyword · More

1 Introduction: Problem Statement

This project aims to develop a fire detection system using a designated dataset. The dataset consists of three subsets: a training set, a validation set, and a test set. Specific constraints and guidelines must be strictly followed to ensure compliance with the project requirements.

1.1 Dataset Access and Constraints

The dataset is available for download at `https://www.kaggle.com/datasets/abdelghaniaaba/wildfire-prediction-dataset/code`. It comprises a training set, a validation set, and a test set. A critical restriction is imposed on the training set: its labels are inaccessible. Any direct utilization of annotations from the training set will lead to disqualification.

1.2 Dataset Partitioning

To facilitate model training, the original validation set must be partitioned into a newly defined validation set and a new training set. The original training set may be leveraged in a creative manner; however, its labels must not be employed under any circumstances.

1.3 Model Development

A deep neural network (DNN) will be trained utilizing the newly defined training and validation sets. Various methodologies and supplementary resources may be incorporated to enhance model performance, provided that all constraints related to annotation usage are rigorously upheld.

2 Method

2.1 Dataset Analysis

. . .

2.2 Proposition 1: Using Available Labeled Data

2.2.1 Naive Coordinates Classifier

...

2.2.2 ResNet Classifier

...

2.3 Self-Supervision (Learning from Unlabeled Data)

2.4 SimCLR

SimCLR (Simple Framework for Contrastive Learning of Visual Representations) is a self-supervised learning method that does not require any labels. It learns visual representations by maximizing the agreement between differently augmented views of the same data example.

The training pipeline of SimCLR consists of the following steps:

2.4.1 Data Augmentation

Each input image is transformed into two different augmented views through a series of random augmentations such as cropping, resizing, flipping, color jittering, and Gaussian blurring. These augmentations are crucial as they create diverse views of the same image, which the model learns to recognize as similar.

2.4.2 Encoder Network

Both augmented views are passed through an encoder network (typically a ResNet) to obtain their respective feature representations. The encoder network is shared between the two views.

2.4.3 Projection Head

The feature representations are then passed through a small neural network called the projection head, which maps them to a latent space where the contrastive loss is applied. The projection head typically consists of a few fully connected layers.

2.4.4 Contrastive Loss

The core idea of SimCLR is to bring the representations of the two augmented views of the same image closer together in the latent space while pushing apart representations of different images. This is achieved using a contrastive loss function, specifically the normalized temperature-scaled cross-entropy loss (NT-Xent loss). The loss function is defined as:

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \tag{1}$$

Where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the latent representations of the two augmented views of the same image, $\mathrm{sim}(\cdot, \cdot)$ denotes the cosine similarity, τ is a temperature parameter, and N is the batch size.

By training the model in this manner, SimCLR learns robust and generalizable visual representations that can be fine-tuned for downstream tasks such as classification, even with limited labeled data.

Just wanted to cite this paper [1].

2.5 Variational Autoencoder (VAE) + Classifier

A Variational Autoencoder (VAE) was employed as a self-supervised learning approach to extract latent representations from the training dataset. The VAE comprises an encoder network, which maps input images to a latent space distribution, and a decoder network, which reconstructs the input from latent variables. The encoder network consists of a series of 4 convolutional layers followed by fully connected layers, while the decoder network mirrors the encoder architecture. The VAE was trained using the unlabeled training dataset. The training objective that we decided to use is Beta-VAE [2]. Beta-VAE is a modification of the original VAE that introduces a hyperparameter β to the loss function. The β parameter controls the trade-off between the reconstruction loss and the Kullback-Leibler divergence. The loss function for the Beta-VAE is given by:

$$\mathcal{L}(\theta, \phi, x, z, \beta) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x) \| p(z)) \tag{2}$$

where $p(z|x)$ is the learned posterior, $p(z)$ is the prior distribution (assumed to be Gaussian), and $p(x|z)$ represents the likelihood of reconstructing the input image from the latent space. Well chosen β values can lead to disentangled representations in the latent space. The VAE was trained for 50 epochs using the Adam optimizer with a learning rate of 0.0001. The latent representations obtained from the VAE were then used to train a classifier. The classifier consists of a two fully connected layer with a ReLU activation fucntion and a Dropout layer in between. The classifier was trained using the labeled train dataset. The training objective was to minimize the binary cross-entropy loss between the predicted and true labels. The linear classifier was trained for 20 epochs using the Adam optimizer with a learning rate of 0.0001.

Table 1: Results of different models with and without pre-training.

Pre-Training	**Model**	**Backbone**	**#params**	**Accuracy**	**F1 Score**
—	Coordinate Classifier	—	—	0.855	0.873
—	Feature Classifier	ResNet18	—	0.983	0.984
		ResNet34	—	—	—
		ResNet50	—	0.985	0.986
		ResNet101	—	—	—
SimCLR	Feature Classifier	ResNet18	—	—	—
		ResNet34	—	—	—
		ResNet50	—	0.976	0.978
		ResNet101	—	—	—
β-VAE	Feature Classifier	ResNet18	—	—	—
		ResNet34	—	—	—
		ResNet50	—	—	—
		ResNet101	—	—	—

3 Conclusion

Your conclusion here

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[2] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.