machine learning & knowledge extraction

MDPI

Article

Multimodal AutoML via Representation Evolution

Blaž Škrlj [1,*,†], Matej Bevec [1,†] and Nada Lavrač [1,2,*]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] School of Engineering and Management, University of Nova Gorica, Glavni trg 8, 5271 Vipava, Slovenia
* Correspondence: blaz.skrlj@ijs.si (B.Š.); nada.lavrac@ijs.si (N.L.)
† These authors contributed equally to this work.

Abstract: With the increasing amounts of available data, learning simultaneously from different types of inputs is becoming necessary to obtain robust and well-performing models. With the advent of representation learning in recent years, lower-dimensional vector-based representations have become available for both images and texts, while automating simultaneous learning from multiple modalities remains a challenging problem. This paper presents an AutoML (automated machine learning) approach to automated machine learning model configuration identification for data composed of two modalities: texts and images. The approach is based on the idea of representation evolution, the process of automatically amplifying heterogeneous representations across several modalities, optimized jointly with a collection of fast, well-regularized linear models. The proposed approach is benchmarked against 11 unimodal and multimodal (texts and images) approaches on four real-life benchmark datasets from different domains. It achieves competitive performance with minimal human effort and low computing requirements, enabling learning from multiple modalities in automated manner for a wider community of researchers.

Keywords: AutoML; representation learning; evolution; multimodal learning

check for updates

Citation: Škrlj, B.; Bevec, M.; Lavrač, N. Multimodal AutoML via Representation Evolution. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1–13. https://doi.org/10.3390/make5010001

Academic Editor: Andreas Holzinger

Received: 2 November 2022
Revised: 1 December 2022
Accepted: 15 December 2022
Published: 23 December 2022

1. Introduction

With the increasing amounts of available computing power, the design, development, and deployment of systems capable of automatic machine learning model configuration identification are becoming a widespread practice. The currently prevailing paradigm studying the design of such systems is that of automated machine learning—the field of AutoML. In recent years, this relatively new field has seen substantial progress; since some of the first AutoML systems, such as *auto-scikit* and similar, extensive efforts have been devoted to better understanding the behaviour of such systems when exposed to real-life scenarios, where different types of data can be simultaneously present alongside background (prior) knowledge, capable of speeding up the subsequent optimization.

Even though many existing approaches already solve the problem of identifying the approximately optimal configuration of a given learning algorithm for tabular data, the development of AutoML systems suitable for operation on heterogeneous data is still a challenge, where the representation of the data most suitable for learning is not necessarily given up-front. The solution proposed in this paper is a scalable multimodal AutoML system, specifically adapted to operate in low-resource settings, with no specialized hardware required. The main contributions are summarized as follows.

1. We propose MuRE (Multimodal Representation Evolution), an AutoML system for low-resource multimodal classification based on the idea of *representation evolution*.
2. The proposed system is evaluated on a collection of four real-life multimodal datasets, which include both text and image-based data.
3. The performance is compared against strong baselines, such as MobileNets and BERT, including fused inputs of the spaces obtained by such models. Qualitative aspects of the final (evolved) representations are also considered.

4. Theoretical properties of MuRE are discussed alongside the current limitations of the approach.

The paper is structured as follows. The related work is presented in Section 2. The proposed MuRE algorithm for multimodal representation evolution is presented in Section 3, followed by its experimental evaluation in Sections 4 and 5. The conclusions and a discussion of the impacts of this work are presented in Section 6.

2. Related Work

In this section, we discuss the notion of *automated learning*—the study of creating systems capable of solving a given task with *minimal* human intervention. The process of machine learning and representation learning, regardless of it being of symbolic or neural nature, is in most cases governed by *hyperparameters*, i.e., *tunable variables* that impact the learning progress/properties [1]. With the increasing availability of computing power, manual tuning of hyperparameters is gradually being overtaken by automatic procedures, i.e., *meta-learning* algorithms. The purpose of this second layer of learning is to automate the redundant and time-consuming manual optimization of a learning algorithm and rely more on the available computing resources—currently, the amount of computing power available to a common user is increasing, even though this might not remain the case [2]. The notion of meta-learning can be understood as learning from *prior* experience in a systematic manner [3,4]. Even though general (naïve) solutions for meta-learning tasks are—due to the no free lunch theorem—practically impossible [5], optimization within subspaces of the relevant solution space can lead to efficient and scalable solutions. For example, when designing a routing search engine, by incorporating the historical data on city-to-city traversals, the algorithm designers do not need to treat all candidate paths as equiprobable and can jump-start the search from the existing solution(s). Furthermore, greedy search is also commonly used to drastically reduce the space size, even if it neglects parts containing reasonable solutions. The development of systems capable of automatic model configuration and data preprocessing has been an active research area in the last few years. We refer to a system capable of automatic model tuning/data configuration as an *AutoML system* [6]. Examples of existing AutoML systems which have already shown promising performances include autoWEKA [6], auto-sklearn [7], and TPOT [8]. Many AutoML systems can be understood as *search* across a non-convex configuration space comprised of configurations/representations.

In recent years, substantial research effort has been focused on designing and optimizing AutoML systems for different domains. Widely used AutoML libraries include for example TPOT [8], OBOE [9], H2O AutoML (https://github.com/h2oai/h2o-3, accessed 25 September 2022), FLAML [10], ML-Plan [11], auto-XGBoost [12], GAMA [13], and others. Even though the early systems focused primarily on tabular data due to the previously available algorithm libraries for this domain, other less structured data sources are being actively explored. Examples include, e.g., automatic exploration of neural network topologies for the task of computer vision [14], graph neural network topologies for relational regression/classification [15]. Furthermore, meta-learning packages built around widely used deep learning libraries, such as Keras, have also gained popularity in recent years (Auto-Keras) [16]. Recently, many novel AutoML methods have been introduced and offered in a form usable to machine learning practitioners. For example, auto-sklearn [7] and autoWEKA [17] (WEKA—Waikato Environment for Knowledge Analysis) are approaches for automatic learning from tabular data. Their goal is to minimize the user's input during hyperparameter tuning and model selection, which they achieve via Bayesian optimization. Further, the process of identifying a suitable deep learning architecture was shown to be suitable for optimization; an example is the NASnet project [18], where large-scale exploration of neural network architectures is conducted automatically.

The field of neural architecture search has grown significantly in recent years [19]. Finally, recent trends indicate that understanding the *transferrability* in the latent space might offer novel and faster ways for neural network model training. An example of

this new paradigm is dataset2vec [20]. Leveraging semantic annotations of systems for better problem-specific learning has also been explored recently. OMA-ML [21] is a recent method that leverages a dedicated ML ontology for guiding the AutoML process itself. It enables exploitation of useful prior knowledge and with it enables faster search, followed by automated generation of reports.

One of the algorithm groups which have stood the test of time is genetic algorithms, which are part of a broader spectrum of methods termed *evolutionary computation*. These algorithms mimic the behaviour of, e.g., cell division/DNA replication and offer a highly parallelizable metaheuristic optimization procedure suitable for most optimization problems. Even though there are no real guarantees regarding their general performance (the no free lunch theorem [5]), they consistently offer a simple-to-implement and efficient automation of many real-life optimization endeavours. Moreover, with the increasing amounts of available computing resources, the relevance of this and similar types of algorithms are gaining traction in the broader machine learning community. This branch of algorithms has been considered since the 1980s [22].

Later developments in this field focus more on multi-objective optimization of the exploration of Pareto fronts, efficient implementations and scalability [23,24]. Their applications are becoming increasingly more relevant due to the larger amounts of available computing resources available. Practical applications include energy management [25], and recent autonomous driving research [26]. Genetic algorithms are the main optimization paradigm considered in this paper.

We finally discuss some of the recent advances in *multimodal* machine learning that impacted this contribution. A recent AutoML approach considered tabular data with intermediary text fields [27]—they applied transformer-based neural networks for feature construction, jointly optimized to achieve human-level competition performance. Further, a recent benchmark that focuses on tabular data, which includes text-based information, was proposed [28] to further the understanding of how AutoML systems perform in such settings. Currently, most solutions considering multimodal data are based on deep learning [29], due to its capacity for representation learning of texts and images. The recent advancements in the field of multimodal AutoML indicate that jointly considering different representation types is a promising research endeavour. This paper builds on these ideas, extending the search to the space of images and texts.

3. MuRE: Multimodal Representation Evolution

We next discuss the proposed *MuRE* approach. We begin by discussing the idea of *representation evolution* (summarized in Figure 1), followed by the formal overview and the description of the final version of the proposed system.

Conceptually, the AutoML we henceforth refer to as MuRE consists of two main conceptual steps: representation learning and configuration search. Both steps are simultaneously considered as part of *evolution*—we adopt the approach commonly referred to as *evolution strategies* [30] due to its compatibility with real-valued inputs (weights of feature spaces). This work builds on the recent implementation of this idea focused exclusively on text-based datasets [31]. The main reason the implementation was considered as the main building block is due to its capability for handling sparse and dense matrices simultaneously, and, further, considering learning algorithms that do not induce substantial memory overheads (which is a common caveat). For example, sparse matrices are common when dealing with symbolic representations (e.g., bags of tokens), however, dense ones are commonplace for document/image embeddings. The minimization problem addressed by the evolution can be in its *general form* stated as follows:

$$\text{Solution} \approx \underset{\substack{\Theta \in \text{algorithmSpace} \\ \times \text{hyperParamSpace} \\ \times \text{transformationSpace}}}{\arg\min} \mathbb{E}\Big[\textsc{Loss}(\textit{learnerClass}, \Theta, \text{data})\Big].$$

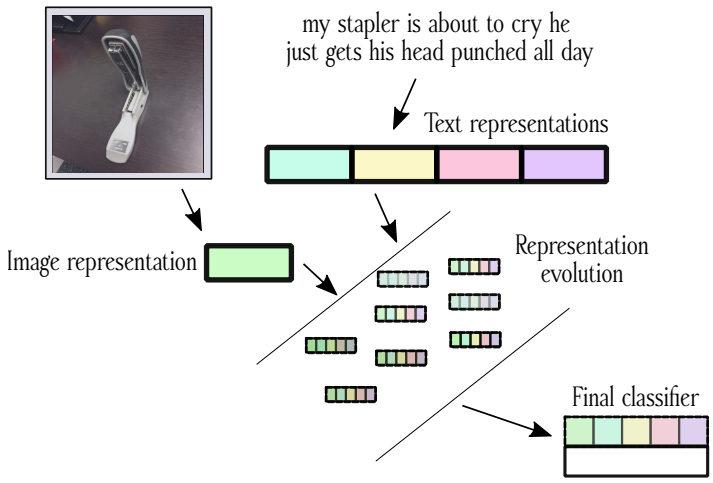

Figure 1. An overview of the proposed MuRE architecture. Different types of inputs (images, texts) are transformed into suitable representations (either embeddings or sparse matrices corresponding to, e.g., n-gram-based features). The representations are used to construct joint representation space that is iteratively re-weighted by evolution strategies. The final result is a re-weighted feature space alongside fine-tuned classifiers (hyperparameters are tuned for each obtained join representation).

The stated problem is a generalization of the one discussed in [6], which considers the space of possible transformations; this space includes, e.g., dimensionality reduction and embedding construction. The proposed **representation evolution** process denotes the process of iterative re-weighting of different data representations and simultaneous evaluation for a given task (e.g., classification). The evolution here is the optimization procedure that enables exploration of differently-amplified representations and the effect of this amplification on the learning performance (fitness function outputs a score based on, e.g., cross validation). Intuitively, MuRE attempts to find the right *balance* between different representations considered, enabling automated prioritization of feature spaces of different types (sparse and dense). Note that the expected value of the loss function for a given configuration Θ is commonly estimated via cross-validation. Even if the stated criterion addresses the construction of the final learner, note that if the LOSS is not representative, the solution obtained as the result of the minimization will not necessarily *generalize*/perform well on unseen data. Note that the LOSS function is task-independent. We next discuss the core components in more detail.

3.1. Representation Learning Phase

The notion of representation learning is becoming of central relevance in many domains of machine learning (and beyond). By deriving machine-readable representations of raw input types (such as texts), subsequent learning can be substantially faster, and the representations can be more easily transferred. The MuRE's initial phase corresponds to the construction of a multitude of different document and image representations, summarized in Table 1. Note that all representation types are automatically considered; however, if a given representation appears as irrelevant, it will be discarded by the method.

Table 1. Representations of documents and images considered by MuRE.

Feature Type	Description	Representation Type
Concept-based features	Features based on triplet groundings	symbolic
Relational tokens	Tokens at a given distance	symbolic
Relational characters	Characters at a given distance	symbolic
Relational bi-grams	Character pairs at a given distance	symbolic
Topics	TF-IDF matrix, transposed and clustered (topics)	symbolic
Keywords	Keyword-based features	symbolic
Words	Word n-grams	symbolic
Characters	Character n-grams	symbolic
Contextual	MPNet-based document embeddings	sub-symbolic
Image embeddings	Obtained with CLIP [32]	sub-symbolic
Document graph	Jaccard-based document graph's node embeddings	sub-symbolic

The reader can observe that both symbolic and sub-symbolic feature types are considered. Consideration of such hybrid spaces results in high-dimensional, mostly sparse feature spaces. All evolution and subsequent learning steps operate with sparse matrices to ensure a low memory footprint.

3.2. Representation Evolution

Once the initial representations are constructed, the performance of a series of linear classifiers is first used to establish the initial real-valued weights of individual spaces—for each of the spaces, a score based on cross-validation is obtained and used as the initial weight. Once all weights are obtained, the evolution proceeds by iteratively perturbing the weights and, for each such perturbation, evaluating a collection of linear classifiers characterized by different loss functions, regularization, and other hyperparameters. The hyperparameters depend on the learning algorithm selected; for linear learners, these hyperparameters include different types of regularizations (L1, L2) and penalty terms (known as alpha parameter). This step is computationally efficient, hence a larger space of classifiers can be explored jointly with each feature weight perturbation. The evolution stores intermediary checkpoints that are useful for either transferring the current set of weights to a new learning scenario or continuing with an existing optimization. The error term considered by stochastic gradient descent is:

$$\textsc{Err}(\boldsymbol{w}, b) = \underbrace{\frac{1}{|D|}\sum_{i=1}^{|D|}\mathcal{L}(\boldsymbol{y}_i, \boldsymbol{w}^T\boldsymbol{x}_i + b)}_{\text{Loss term}} + \alpha\left[\underbrace{\frac{1-\beta}{2}\sum_{i=1}^{|D|}\boldsymbol{w}_i^2}_{\text{L2}} + \underbrace{\beta\sum_{i=1}^{|D|}|\boldsymbol{w}_i|}_{\text{L1}}\right],$$

where $\boldsymbol{y}$ is the target vector, $\boldsymbol{x}_i$ the i-th instance, $\boldsymbol{w}$ is a weight vector, $\mathcal{L}$ is the considered loss function, and α and β are two numeric hyperparameters: α represents the overall weight of the regularization term, and β the ratio between L1 and L2. The loss functions considered are the hinge and the log loss. In the final stage of optimization, an extended space of classifiers is explored (in parallel), to obtain the final representation-learner configuration. Each such configuration can be stored as a compressed object, making the outputs of MuRE runs easily transferable. Current implementation of MuRE supports different evaluation regimes; the default one is stratified cross validation (fivefold). Furthermore, the weights corresponding to individual feature spaces are real valued vectors that are *transferable* between different learning tasks. This way, a single, longer AutoML run can provide valuable initial conditions for subsequent runs, and, as such, reduce the number of generations required to obtain feasible solutions.

4. Experiments

This section discusses the experimental setting used to evaluate MuRE's performance on real-life multimodal datasets against strong (and weak) baselines.

4.1. Datasets

We evaluate the performance of our approach on the task of multimodal classification from images and associated texts on five different datasets. A short overview of the considered datasets is provided in Table 2. *Tasty Recipes* [33] is a small collection of recipes, represented by a textual document, including the ingredient list and preparation instructions, and an image of the described dish. Each recipe is classified as one of 25 food categories, such as "tacos". *Caltech Birds* [34] is a multimodal extension of *Caltech-UCSD Birds (CUB-200)* [35] a popular fine-grained image classification dataset with images of birds belonging to one of 200 often visually similar species. Each image is augmented with textual descriptions of the given bird's physical features provided by human annotators in this dataset.

In the above cases, data from different modalities can be seen as additional information about instances that can help improve prediction. *Fauxtography* [36] and *Fakeddit* [37] datasets, however, entail the task of *fact-checking*, which is inherently multimodal. Here, an image–text pair can be valid, meaning the image and text convey real and matching information. Conversely, a pair can be invalid, meaning either the image or the text are false, misleading or manipulated, or both the image and the text are real but mismatched, usually when a description makes a false claim about a real image. Borrowing its name from the practice of manipulating photographs in order to deceive, *Fauxtography* depicts scenes from world news gathered from Snopes, a fact-checking website with a collection of both true and deliberately false news articles. The dataset is balanced with additional true image-text pairs coming from the *Reuters' Pictures of the year* segment. *Fakeddit* is collected from specific Reddit communities (called *subreddits*), where, for a given instance, its source subreddit determines its class. It offers binary, 3-way, and 6-way target variables, of which we only consider the first—predicting whether an image–text pair is true (real) or false (fake). Fakeddit is originally a very large dataset (1M examples). However, we take a 5 k sub-sample to approximately match the scale of Fauxtography since we focus primarily on small datasets. We call this dataset *Fakeddit 5k*.

All considered datasets are organized as follows. Each training example is represented by one image, one textual document, and the associated target variable. We make these datasets available in the described consistent format along with the provided code.

Table 2. Overview of the used multimodal datasets, including their size, task, and a description of the data.

Dataset	#Instances	Task	Data	Target
Tasty Recipes [33]	271	multiclass classification	Textual recipes and images of the described dishes	25 food categories
Fauxtography [36]	1354	binary classification	Images and descriptions of world news	True if image-text pair is factual and matching, False otherwise.
Fakeddit 5 k [37]	4880	binary classification	Titles and images associated with Reddit posts from various communities ("subreddits")	True if image-text pair is factual and matching, False otherwise.
Caltech Birds [34]	11788	multiclass classification	Images of various bird species and descriptions of their physical features	192 bird species

4.2. Baselines and Evaluation

Our work encompasses machine learning with images and text, multimodal learning, as well as AutoML. As such, we conduct experiments to compare our method to image-only baselines, text-only baselines, including an AutoML approach and multimodal baselines. Since the focus of our approach is quick deployment (prototyping) on a new prediction problem with consumer-grade hardware and by users who may or may not be

domain experts, most baselines rely heavily on transfer learning with pre-trained models. Hyperparameters used are described below. No automated hyperparameter optimization (e.g., grid search) was performed. The described baseline methods are implemented using scikit-learn [38], and pytorch [39].

4.2.1. Image-Only Baselines

The following classification approaches that only consider the input image are tested:

- **MobileNet + SVM**
  Outputs from the second-to-last layer of a pre-trained MobileNetV3 [40] model are taken as image features and fed into a linear SVM for classification. Specifically, we use the MobileNet V3 Large architecture, pretrained on ImageNet. We choose the same configuration whenever MobileNet is utilized. The SVM classifier is trained using *hingle loss* for a maximum of 10,000 iterations with regularization constant $C = 2$ and other parameters equal to scikit-learn defaults. (https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html, accessed on 10 November 2022).
- **MobileNet + NN**
  Outputs of the second-to-last layer of a pre-trained MobileNetV3 model are fed into a two-layer fully-connected network for classification. The neural classifier uses a 100-dimensional hidden layer and ReLU activation. It is trained for 30 epochs at a base learning rate of 10^{-3} using the Adam optimizer [41] with an L2 penalty of 10^{-4}.
- **Fine-tuned MobileNet**
  The final classification layer of a pre-trained MobileNetV3 is replaced to conform to the desired output (i.e., a single fully-connected layer). The whole network is then fine-tuned for 20 epochs, at a base learning rate of 5×10^{-4} using the Adam optimizer with an L2 penalty of 10^{-4}.

We chose to base our experiments on the MobileNet architecture [40] since it achieves comparable performance to other larger state-of-the-art models, but it is significantly smaller and faster. This falls in line with the setting of our system, which should be quickly deployable on a consumer-grade machine. However, MobileNet could be substituted with other architectures, such as ResNet [42] or EfficientNet [14], for marginal increases in performance at a significant cost of computational load (The provided experimental code includes experiments to demonstrate this).

4.2.2. Text-Only Baselines

The following classification approaches that only consider the input text are tested:

- **N-grams + SVM**
  Word and character n-grams are TF-IDF-vectorized and fed into a linear SVM for classification. Nine different regularization values of C (from 0.1 to 500) are tested when training, with the best performing model being chosen for the final prediction. Maximum iterations are set to 100,000 and other parameters equal to scikit-learn defaults. (https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html, accessed on 10 November 2022) This baseline represents a "traditional" text classification approach.
- **Fine-tuned BERT**
  A pre-trained BERT language model, configured as a classifier, is fine-tuned on our data. We use the base uncased pretrained model, trained on an English language corpus using an MLM objective [43]. The fine-tuning is performed for 20 epochs using the AdamW optimizer [44] at a base learning rate of 4×10^{-5}.
- **TPOT**
  TPOT [8,45] is an easy-to-use AutoML tool that automatically evolves scikit-learn pipelines based on tree ensemble learners to optimize performance on a classification task. We deploy TPOT on a vectorized word n-gram space with default settings (http://epistasislab.github.io/tpot/api/, accessed on 1 November 2022).

4.2.3. Multimodal Baselines

We also consider several common multimodal approaches that combine information from images and texts using both early fusion and late fusion.

- **Late fusion (sum)**
  An image-only classifier (MobileNet + SVM) and a text-only classifier (N-grams + SVM), both as described above, are trained independently. Class probability distributions are extracted from both models and combined by a sum. Combined probability for class i is defined as:

$$P(i) = softmax(P_{img}(i) + P_{txt}(i))$$

- **Late fusion (max)**
  Class probabilities are obtained from both models and combined by a maximum. Combined probability for class i is defined as:

$$P(i) = softmax(max(P_{img}(i), P_{txt}(i)))$$

- **Early fusion + SVM**
  Image features are extracted with a MobileNetV3 model, much like in Section 4.2.1. Text features are extracted using a pre-trained Sentence-BERT sentence embedding model [46], trained using a multilingual MPNet objective [47]. Features from both modalities are then concatenated and fed into a linear SVM for prediction. The SVM classifier is trained for a maximum of 1000 iterations with $C = 2$.
- **Early fusion + NN**
  Again, image features are extracted with a MobileNetV3 model and text features are extracted using a Sentence-BERT model. Features from both modalities are then concatenated and fed into a two-layer fully-connected network. The neural classifier uses a 100-dimensional hidden layer and ReLU activation and is trained for 30 epochs, at a base learning rate of 10^{-3} using the Adam optimizer with an L2 penalty of 10^{-4}.

We consider a MuRE baseline, given at most one hour for training. The variant adopts minority class upsampling (addition of artificial instances up to the point of uniform distribution of labels). We hypothesized that by considering upsampling, the MuRE's classifier layer (linear) could perform better.

5. Results

This section presents the classification performance results across different considered datasets.

5.1. Quantitative Results

The overview of the results is shown in Table 3.

Overall, the results can be summarized as follows. The proposed MuRE performs on average the best (3.25 average rank), followed by BERT (4.5) and late fusion via summation (4.5). The early fusion did not perform as well (ranked 5th on average). The results indicate that multimodal approaches dominate, which indicates that the considered problems are indeed best solvable by considering both modalities. The proposed MuRE performs subpar on fakeddit dataset, however, is very competitive on other datasets. It outperforms all other approaches on caltech-birds dataset; this dataset requires that approaches learn from very few instances per label, indicating MuRE's data efficiency. Another observation is that multimodal models always outperform image-only models. There is no clear winner in terms of considering early or late fusion. Overall, AutoML approaches are on average ranked high. Detailed benchmark is specified as part of the Supplementary Material. Furthermore, results also indicate sub-optimal performance on the Fakeddit (I+T) dataset. The best-performing method for this dataset was a text-only BERT-based model. For this task, the follow-up results were obtained by considering early fusion

(combining embeddings). This result indicates that a focused combination of sub-symbolic-only representations can offer competitive results. The worse performance of MuRE on this task can be attributed to potential noise introduced by considering high-dimensional sparse spaces alongside sub-symbolic ones; it is possible that the symbolic part of the space did not encode the relevant information present in embeddings, and, thus, impacted the learning in a negative manner. The task itself is defined as binary classification of whether a given text+image combination is *factual*. Given that MuRE considers also knowledge-graph-based representations, it is possible this part of information, due to imperfect mapping, offers incomplete information (note that only text titles are given, which might not have been enough in this case). Should the best-performing (for this task) BERT model capture these relations better, this is a possible explanation for better performance.

Table 3. Classification results—10 fold cross validation, same seed for all methods (420). Green cells denote either first or second rank for a given dataset (I = images, T = texts).

Metric	**Macro F1**				**Accuracy**				**Precision**				**Recall**			
Approach/Data Set	**Recipes**	**Faux**	**Fake**	**Birds**	**Recipes**	**Faux**	**Fake**	**Birds**	**Recipes**	**Faux**	**Fake**	**Birds**	**Recipes**	**Faux**	**Fake**	**Birds**
Majority classifier	0.002	0.354	0.378	0.0	0.03	0.547	0.608	0.002	0.001	0.274	0.304	0.0	0.042	0.5	0.5	0.005
(I) MobileNet + SVM	0.623	0.683	0.712	0.699	0.671	0.685	0.726	0.702	0.667	0.684	0.713	0.708	0.646	0.684	0.713	0.707
(I) MobileNet + NN	0.423	0.745	0.742	0.692	0.476	0.748	0.754	0.694	0.471	0.747	0.742	0.707	0.473	0.745	0.742	0.7
(I) Fine-tuned MobileNet	0.512	0.354	0.378	0.192	0.568	0.547	0.608	0.199	0.567	0.274	0.304	0.23	0.554	0.5	0.5	0.202
(T) N-grams + SVM	0.852	0.801	0.753	0.602	0.875	0.805	0.768	0.618	0.872	0.807	0.758	0.611	0.889	0.799	0.75	0.624
(T) Fine-tuned BERT	0.777	0.813	0.829	0.593	0.804	0.818	0.836	0.595	0.814	0.825	0.828	0.613	0.819	0.811	0.83	0.597
(T) TPOT	0.861	0.791	0.732	0.612	0.889	0.803	0.75	0.617	0.877	0.832	0.739	0.632	0.889	0.788	0.729	0.622
(I+T) Late fusion (sum)	0.79	0.76	0.773	0.781	0.822	0.764	0.786	0.782	0.813	0.765	0.777	0.789	0.818	0.758	0.771	0.787
(I+T) Late fusion (max)	0.73	0.76	0.773	0.762	0.771	0.764	0.786	0.764	0.761	0.765	0.777	0.771	0.763	0.758	0.771	0.769
(I+T) Late fusion (stacking)	0.711	0.72	0.781	0.768	0.756	0.722	0.792	0.768	0.751	0.72	0.783	0.779	0.739	0.72	0.78	0.773
(I+T) Early fusion + NN	0.5	0.796	0.82	0.694	0.568	0.801	0.828	0.697	0.534	0.806	0.82	0.704	0.568	0.794	0.821	0.701
(I+T) Early fusion + SVM	0.746	0.784	0.816	0.721	0.808	0.788	0.824	0.726	0.782	0.789	0.816	0.728	0.773	0.783	0.817	0.732
(I+T) MuRE-1h	0.913	0.812	0.766	0.911	0.952	0.817	0.776	0.948	0.911	0.824	0.773	0.911	0.923	0.813	0.773	0.919

The overall results indicate that automated representation learning serves as a strong baseline against strong competitors based on single modalities. Furthermore, it outperforms other commonly adopted fusion approaches, making it a strong baseline for multimodal tasks considered. To better understand whether the multimodal representation space already reflects the class structure, we further visualized the 2D projection [48] of the high-dimensional space, colored by the class assignments. The result is shown in Figure 2.

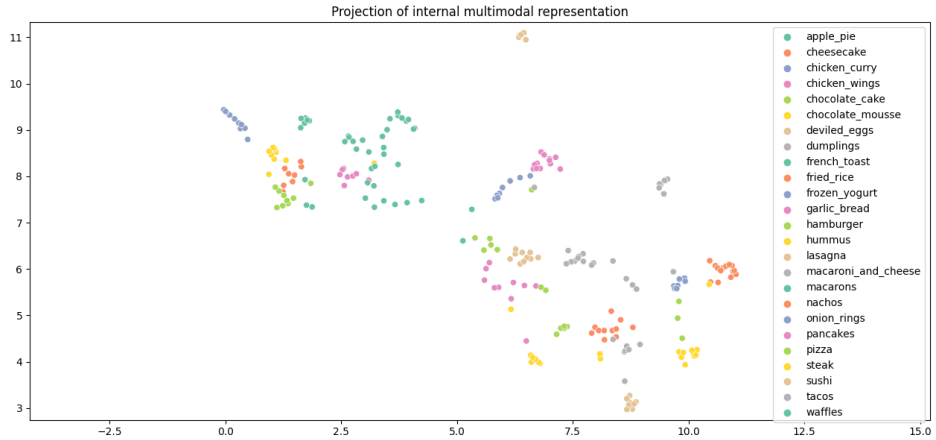

Figure 2. Visualization of the projection of multimodal latent space (I+T) for the *tasty* dataset.

The visualization demonstrates that the joint representation space already segments the space of instances according to class assignments. Even though some of the classes are less well separated, relatively many denser clusters emerge.

5.2. Ablation Study—Subspace Importances

The proposed MuRE approach builds on the idea of *representation evolution*, the process of iteratively re-weighting differently-typed feature subspaces. The results can be interpreted—the larger the subspace weight, the more apparent its impact on the linear learning layer considered for final classification. Visualized importances for *tasty* dataset are shown in Figure 3.

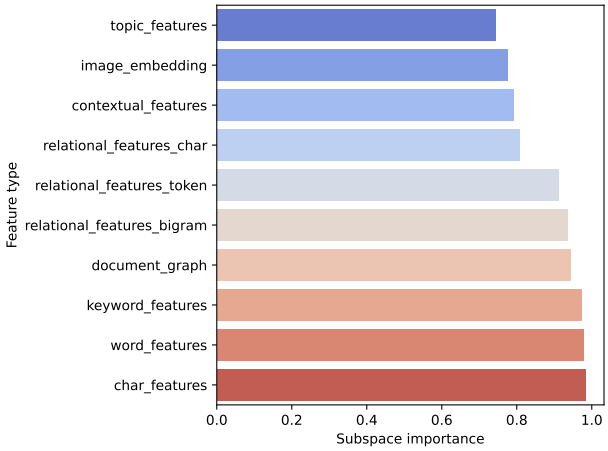

Figure 3. Weights of different feature subspaces—*tasty*. For this dataset, image-only classifiers perform poorly—this is also apparent in the joint space considered by MuRE; here, image embeddings have low final weights relative to text-based ones. Character and token features are the most relevant for this problem (text-only learning is a strong baseline).

Figure demonstrates MuRE's capability to offer not only automated learning from multi-modal inputs, but also some degree of *explainability*. This type of ablation is part of the automatically generated report, and, as such, enables the machine learning practitioner to directly inspect the impact of different data types, but also detect potential issues with them early on (before spending more time on modeling). The visualizations are generated automatically with regards to feature subspaces considered for learning.

The weights can be transferred between the experiments—this way, task transfer is possible. Apart from being able to transfer weights and use them as priors for subsequent runs, this space can also be used to understand why different tasks group together/are similarly hard to solve.

6. Conclusions

In this paper, we proposed an AutoML system for multimodal classification; the considered modalities were texts and images. We first demonstrated that both types of inputs, when considered from a representation learning perspective, function as a part of *representation evolution*, the idea of iterative representation refinement at a task level. We demonstrated that representation evolution performs on-par with strong baselines that consider a single modality but also multimodal baselines. As such, MuRE offers an easy-to-implement baseline that is nontrivial to beat, expanding the realm of multimodal learning to practitioners not well versed in this field. Furthermore, the proposed method does not require any specialized hardware, and is, as such, suitable for performing experiments on personal machines. This paper also offers an extensive comparison of the behaviour of multimodal vs unimodal methods. The results indicate that multimodal methods on average perform better. Further, the dependency of a method's success on either of the modalities considered is dataset dependent, even though unimodal solutions perform, on average, worse. Identifying the correct ratio between the extents to which either of the modalities is considered is a challenging problem; however, based on the current results, solvable via representation evolution.

As part of further work, we will extend the MuRE to operate with other modalities, including sound, knowledge graphs (subject–predicate–object triplets) and similar relational data. Furthermore, a sensible research direction includes the study of meta transfer across different modalities—does the relevance of images translate across tasks? Can such information be used as prior knowledge to initialize subsequent optimizations? Finally, we plan to test the approach on the newly introduced shared tasks to evaluate whether it performs on the human levels of task solving. Finally, comparing MuRE's performance against computationally more intense baselines, such as end-to-end CLIP [32], is a sensible research direction.

The proposed MuRE substantially lowers the knowledge required by a machine learning practitioner/data scientist to inspect multimodal learning scenarios. It was built to serve as a strong baseline that can be considered with a few lines of code, albeit offering hard-to-beat performance. By demonstrating that multimodal inputs are suitable also for AutoML-based learning, we believe multiple interesting applications of the tool to novel datasets are possible.

Supplementary Materials: The following are available online at https://github.com/MatejBevec/mmlearn.

Author Contributions: Conceptualization, M.B.; software, B.Š.; formal analysis, M.B.; data curation, B.Š. and M.B.; writing—original draft preparation, B.Š. and M.B.; writing—review and editing, N.L.; supervision, N.L.; funding acquisition, N.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreement 863059 (FNS-Cloud, Food Nutrition Security). The work was also supported by the Slovenian Research Agency (ARRS) core research programme Knowledge Technologies (P2-0103), and projects Computer-assisted multilingual news discourse analysis with contextual embeddings (J6-2581) and Quantitative and qualitative analysis of the unregulated corporate financial reporting (J5-2554). The work was also supported by the Ministry of Culture of Republic of Slovenia through project Development of Slovene in Digital Environment (RSDO).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The main MuRE idea will be relased as part of autoBOT framework for AutoML—https://github.com/SkBlaz/autobot, accessed on 12 December 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, X.; Zhao, K.; Chu, X. AutoML: A Survey of the State-of-the-Art. *Knowl.-Based Syst.* **2021**, *212*, 106622. [CrossRef]
2. Theis, T.N.; Wong, H.S.P. The end of moore's law: A new beginning for information technology. *Comput. Sci. Eng.* **2017**, *19*, 41–50. [CrossRef]
3. Lemke, C.; Budka, M.; Gabrys, B. Metalearning: A survey of trends and technologies. *Artif. Intell. Rev.* **2015**, *44*, 117–130. [CrossRef] [PubMed]
4. Hutter, F.; Kotthoff, L.; Vanschoren, J. (Eds.) Meta-Learning. In *Automated Machine Learning: Methods, Systems, Challenges*; Springer International Publishing: Cham, Switzerland, 2019; pp. 35–61. [CrossRef]
5. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
6. Kotthoff, L.; Thornton, C.; Hoos, H.H.; Hutter, F.; Leyton-Brown, K. Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 81–95.
7. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.T.; Blum, M.; Hutter, F. Auto-sklearn: Efficient and robust automated machine learning. In *Automated Machine Learning*; Springer: Cham, Switzerland, 2019; pp. 113–134.
8. Olson, R.S.; Moore, J.H. TPOT: A tree-based pipeline optimization tool for automating machine learning. In Proceedings of the Workshop on Automatic Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 66–74.
9. Yang, C.; Akimoto, Y.; Kim, D.W.; Udell, M. OBOE: Collaborative filtering for AutoML model selection. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1173–1183.

10. Wang, C.; Wu, Q.; Weimer, M.; Zhu, E.E. FLAML: A Fast and Lightweight AutoML Library. In Proceedings of the 4th Conference on Machine Learning and Systems (MLSys 2021), San Jose, CA, USA, 4–7 April 2021.
11. Mohr, F.; Wever, M.; Hüllermeier, E. ML-Plan: Automated machine learning via hierarchical planning. *Mach. Learn.* **2018**, *107*, 1495–1515. . [CrossRef]
12. Thomas, J.; Coors, S.; Bischl, B. Automatic Gradient Boosting. In Proceedings of the International Workshop on Automatic Machine Learning at ICML, Stockholm, Sweden, 14 July 2018.
13. Gijsbers, P.; Vanschoren, J. GAMA: A General Automated Machine learning Assistant. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bilbao, Spain, 13 September 2021.
14. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
15. Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.B.; Leiserson, C.E. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
16. Jin, H.; Song, Q.; Hu, X. Auto-Keras: An efficient neural architecture search system. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1946–1956.
17. Thornton, C.; Hutter, F.; Hoos, H.H.; Leyton-Brown, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 847–855.
18. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. *arXiv* **2018**, arXiv:1707.07012.
19. Elsken, T.; Staffler, B.; Metzen, J.H.; Hutter, F. Meta-Learning of Neural Architectures for Few-Shot Learning. *arXiv* **2020**, arXiv:1911.11090.
20. Jomaa, H.S.; Schmidt-Thieme, L.; Grabocka, J. Dataset2vec: Learning dataset meta-features. *Data Min. Knowl. Discov.* **2021**, *35*, 964–985. [CrossRef]
21. Humm, B.G.; Zender, A. An ontology-based concept for meta automl. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Hersonissos, Greece, 17–20 June 2021; pp. 117–128.
22. Davis, L. *Handbook of Genetic Algorithms*, 1st ed.; Van Nostrand Reinhold: Washington, DC, USA, 1991.
23. Doerr, B.; Le, H.P.; Makhmara, R.; Nguyen, T.D. Fast genetic algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 15–19 July 2017; pp. 777–784.
24. Corus, D.; Oliveto, P.S. Standard steady state genetic algorithms can hillclimb faster than mutation-only evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2017**, *22*, 720–732. [CrossRef]
25. Leonori, S.; Paschero, M.; Mascioli, F.M.F.; Rizzi, A. Optimization strategies for Microgrid energy management systems by Genetic Algorithms. *Appl. Soft Comput.* **2020**, *86*, 105903. [CrossRef]
26. Li, D.; Deng, L.; Cai, Z. Intelligent vehicle network system and smart city management based on genetic algorithms and image perception. *Mech. Syst. Signal Process.* **2020**, *141*, 106623. [CrossRef]
27. Shi, X.; Mueller, J.; Erickson, N.; Li, M.; Smola, A. Multimodal AutoML on Structured Tables with Text Fields. In Proceedings of the 8th ICML Workshop on Automated Machine Learning (AutoML), Virtual, 23 July 2021.
28. Shi, X.; Mueller, J.; Erickson, N.; Li, M.; Smola, A.J. Benchmarking Multimodal AutoML for Tabular Data with Text Fields. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, Online, 6–14 December 2021.
29. Ramachandram, D.; Taylor, G.W. Deep Multimodal Learning: A Survey on Recent Advances and Trends. *IEEE Signal Process. Mag.* **2017**, *34*, 96–108. . [CrossRef]
30. Beyer, H.G.; Schwefel, H.P. Evolution strategies—A comprehensive introduction. *Nat. Comput.* **2002**, *1*, 3–52. [CrossRef]
31. Škrlj, B.; Martinc, M.; Lavrač, N.; Pollak, S. autoBOT: Evolving neuro-symbolic representations for explainable low resource text classification. *Mach. Learn.* **2021**, *110*, 989–1028. [CrossRef]
32. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–14 August 2021; pp. 8748–8763.
33. Kaplenko, M. Multimodal Classification, 2019. Available online: https://github.com/xkaple01/multimodal-classification (accessed on 20 November 2022).
34. Reed, S.; Akata, Z.; Lee, H.; Schiele, B. Learning deep representations of fine-grained visual descriptions. In Proceedings of the IEEE Conference onComputer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 49–58.
35. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report CNS-TR-2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
36. Zlatkova, D.; Nakov, P.; Koychev, I. Fact-checking meets fauxtography: Verifying claims about images. *arXiv* **2019**, arXiv:1908.11722.
37. Nakamura, K.; Levy, S.; Wang, W.Y. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv* **2019**, arXiv:1911.03854.
38. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

39. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
40. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference onComputer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
44. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
45. Le, T.T.; Fu, W.; Moore, J.H. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics* **2020**, *36*, 250–256. [CrossRef] [PubMed]
46. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
47. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.Y. Mpnet: Masked and permuted pre-training for language understanding. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 16857–16867.
48. McInnes, L.; Healy, J.; Saul, N.; Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **2018**, *3*, 861. [CrossRef]