Preprint version; final paper published in:
ISPRS J. Photogramm. Remote Sens., 204, 275-290, 2023, https://doi.org/10.1016/j.isprsjprs.2023.09.001

Deep multitask learning with label interdependency distillation for multicriteria street-level image classification

Patrick Aravena Pelizari[a,b,*], Christian Geiß[a,c], Sandro Groth[a], Hannes Taubenböck[a,b]

[a] *German Aerospace Center (DLR), German Remote Sensing Data Center (DFD), Oberpfaffenhofen, 82234 Weßling, Germany*

[b] *University of Würzburg, Institute of Geography and Geology, Department of Remote Sensing, 97074 Würzburg, Germany*

[c] *University of Bonn, Department of Geography, 53115 Bonn, Germany*

* Corresponding author (e-mail address: patrick.aravenapelizari@dlr.de)

Keywords:
Image classification
Multitask learning
Intermediate prediction
Label interdependencies
Street-level imagery
Building characterization

Abstract

Multitask learning (MTL) aims at beneficial joint solving of multiple prediction problems by sharing information across different tasks. However, without adequate consideration of interdependencies, MTL models are prone to miss valuable information. In this paper, we introduce a novel deep MTL architecture that specifically encodes cross-task interdependencies within the setting of multiple image classification problems. Based on task-wise interim class label probability predictions by an intermediately supervised hard parameter sharing convolutional neural network, interdependencies are inferred in two ways: *i)* by directly stacking label probability sequences to the image feature vector (*i.e.*, *multitask stacking*), and *ii)* by passing probability sequences to gated recurrent unit-based recurrent neural networks to explicitly learn cross-task interdependency representations and stacking those to the image feature vector (*i.e.*, *interdependency representation learning*). The proposed MTL architecture is applied as a tool for generic multi-criteria building characterization using street-level imagery related to risk assessments toward multiple natural hazards. Experimental results for classifying buildings according to five vulnerability-related target variables (*i.e.,* five learning tasks), namely *height*, *lateral load-resisting system material*, *seismic building structural type*, *roof shape,* and *block position* are obtained for the Chilean capital Santiago de Chile. Our MTL methods with cross-task label interdependency modeling consistently outperform single task learning (STL) and classical hard parameter sharing MTL alike. Even when starting already from high classification accuracy levels, estimated generalization capabilities can be further improved by considerable margins of accumulated task-specific residuals beyond +6% κ. Thereby, the combination of *multitask stacking* and *interdependency representation learning* attains the highest accuracy estimates for the addressed task and data setting (up to cross-task accuracy mean values of 88.43% overall accuracy and 84.49% κ). From an efficiency perspective, the proposed MTL methods turn out to be substantially favorable compared to STL in terms of training time consumption.

1. Introduction

Geotagged imaging sensor data are an essential source for obtaining spatial information in an automated fashion. This has been greatly promoted by the ever-increasing availability of data collection initiatives (*remote* and *in-situ* sensing) and social media as well as consecutively improving data analysis methods – particularly in the field of artificial intelligence (Ibrahim et al., 2020). Image classification, *i.e.*, the correct labeling of images according to their content into predefined discrete semantic categories, is an important task in this context (Cheng et al., 2020; Biljecki and Ito, 2021). In remote sensing, this task is widely referred to as scene classification and has been deployed particularly for Land Use/Land Cover analysis in very high-resolution data on a variety of thematic foci (Cheng et al., 2020).

Recently, street-level images (SLI), *i.e.*, geotagged photos taken *in-situ* along street courses have shown relevance for a plethora of geospatial applications (Biljecki and Ito, 2021). Capturing the streetscape profile from a human vision perspective with a high level of detail, SLI provide a rich and complementary counterpart to the synoptic view provided by remote sensing (Zhang et al., 2019a; Chen et al., 2022). SLI are widely accessible via global web mapping services such as Mapillary or Google Street View (GSV; Anguelov et al., 2010) but can also be obtained from image-hosting social media platforms such as Flickr (Hoffmann et al., 2023). Furthermore, SLI is collected and disseminated in the framework of local studies (Wieland et al., 2012; Geiß et al., 2017a; Esquivel-Salas et al., 2022).

This study deals with the supervised classification of SLI for information extraction (here: on buildings). Such techniques foresee assigning a thematic label given a limited amount of properly encoded prior knowledge, *i.e.*, labeled training data. The training data is deployed to infer a prediction model, which aims to accurately generalize for unseen, *i.e.*, unlabeled, instances (Geiß et al., 2019). Due to their great advances in solving perceptual tasks in the image domain, recent studies on information gathering with image classification particularly relied on deep learning (LeCun et al., 2015; Cheng et al., 2020), which is also in the scope here.

1.1. Application focus: Building characterization for natural hazard risk assessments

When assessing natural hazard risk, an up-to-date model of the exposed built environment is a critical input (Geiß and Taubenböck, 2013). Such a model needs to cover the spatially allocated exposed assets each assigned with a set of attributes relevant to characterizing their vulnerability to the considered hazards (Taubenböck et al., 2009; Pittore et al., 2017; Gomez-Zapata et. al, 2022). Specific taxonomies are employed for this concern, among them the GED4ALL multi-hazard building classification system proposed by Silva et al. (2022) covers, *e.g.*: the *lateral load-resisting system (LLRS*; *i.e.,* the structural system that resists acting lateral forces such as seismic loads, wind loads, water pressure or earth pressure) and its *material* (*e.g.*, masonry or wood), *height*, *occupancy*, *shape of the building plan*, *structural irregularity*, and *roof shape* among others.

The extraction of building attributes that are relevant in this context using deep learning-based SLI classification methods has been the subject of several recent studies, *e.g.*: Kang et al. (2018) and Hoffmann et al. (2023) assign land-use classes to buildings, and Sun et al. (2022) derive building age and architectural style epoch, respectively. Centered on the topic of building vulnerability, *e.g.*, Gonzalez et al. (2020) derive buildings' *LLRS*, its *material type*, and

ductility, Aravena Pelizari et al. (2021) assess the potential to derive *LLRS material* types along with the building *height* independently as well as their combination, *i.e.*, *seismic building structural types*; Yu et al. (2020) identify vulnerable *soft-storey* buildings and Rueda-Plata et al. (2021) classify unreinforced masonry buildings according to their *roof diaphragm flexibility*. Generalization accuracies have shown that the combined use of SLI and deep learning image classification techniques yields a high potential for the automated complementation of inventory databases, providing an alternative to costly and labor-intensive field surveys or manual annotation campaigns.

The mentioned studies use an individual model for inferring a particular target variable and when targeting the vulnerability of buildings, they focus on a single hazard only (*i.e.*, just earthquakes). Here, in contrast, we provide a methodology for a more comprehensive and thus more generic building characterization. This is particularly relevant in multihazard risk assessments, where multiple different building characteristics may determine the vulnerability to different natural hazards (Pittore et al., 2017; Silva et al., 2022). Thereby, employing *multitask learning* (MTL), we beneficially encode prevailing relationships among building attributes. Striking examples are, *e.g.*, the interdependencies between building age and architectural style (Sun et al., 2022) or between construction practices and building height – due to statically constraints, it is very unlikely that a masonry building has more than 5 storeys (Santa María et al., 2017). In this paper, we propose a deep MTL framework that explicitly accounts for such interdependencies among multiple target variables in an adaptive way. This considerably improves the accuracy of building characterization with SLI. Its application is demonstrated and evaluated based on a comprehensive reference data set, which was annotated according to five vulnerability-related target variables, *i.e.*: building *height*, *LLRS material type*, *SBST*, *roof shape*, and the *block position* (Fig. 1). The latter refers to the position of a building or housing entity in relation to its neighboring buildings or housing entities.

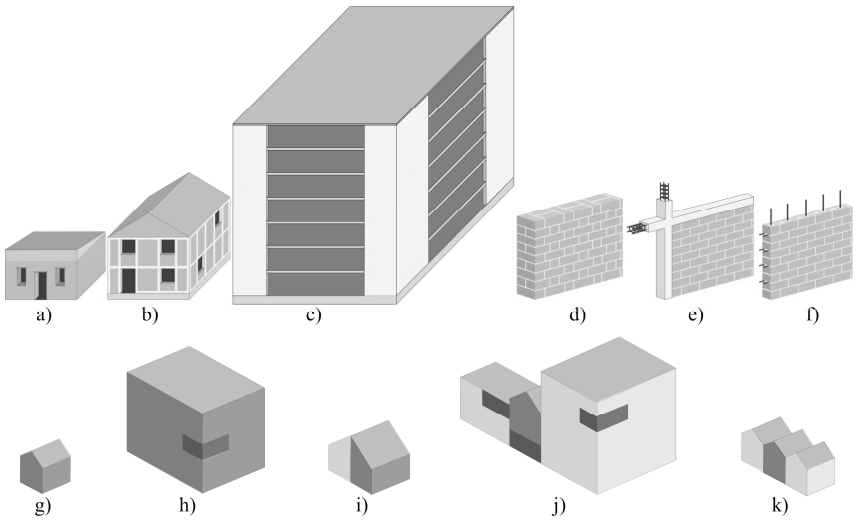


Fig. 1. Schematic exemplification of addressed building characteristics (a – c: height, LLRS material, seismic building structural type, roof shape; d – f: details on masonry LLRSs; g – k: block position): a) 1 storey, unreinforced masonry, MUR_H1, monopitch roof; b) 2 storey, confined masonry, MCF_1-2, pitched or gabled roof; c) 5-7 storeys, reinforced concrete, CR_5-7; d) unreinforced masonry wall; e) confined masonry wall, *i.e.*, masonry with reinforced concrete confinement; f) reinforced masonry wall, *i.e.*, masonry with steel bar reinforcement; g) detached single-party; h) detached multi-party; i) semi-detached; j) adjoining block development; k) adjoining terraced.

Regardless of the application, the presented MTL framework is generic w.r.t. input image data and addressed classification tasks. Thus, it is applicable under arbitrary multitask image classification settings, potentially allowing to improve classification performances due to the distillation of occurring label interdependencies (*e.g.*, to beneficially solve multiple scene classification tasks in aerial or satellite remote sensing data).

1.2. Deep multitask learning

For many real-world problems, it is not feasible to obtain an extensive amount of representative training instances to fully exploit the accuracy capabilities of the deployed classification algorithms. Such settings can benefit from inductive knowledge transfer among multiple related target domains (Pan and Yang, 2010; Zhang and Yang, 2022). If the application requires the inference of multiple target variables MTL is a promising approach. Instead of addressing multiple target outputs each within an independent learning task (*i.e.*, *single task learning*; STL), MTL algorithms learn numerous tasks jointly. Thereby, the governing paradigm is to leverage the intrinsic domain-specific information of several related tasks to improve the generalization performance of all the tasks (Caruana, 1997). In this sense, MTL can exploit synergies among tasks and acts as a regularizer by introducing an inductive bias that reduces the risk of overfitting (Pan and Yang, 2010; Ruder et al., 2017). Hence, particularly settings with multiple related tasks but limited training data can benefit from MTL (Long et al., 2017; Liu and Shi, 2020; Zheng et al., 2022).

In deep learning, MTL involves the joint learning of shared representations from multiple associated supervisory signals. The sharing of layers can substantially decrease training and inference times compared to STL, increasing overall efficiency. This yields the potential to alleviate the common challenges of deep learning concerning data requirements and computational demands. Deep MTL methods typically employ either hard or soft parameter sharing of hidden layers to exploit complementary knowledge among tasks (Ruder et al., 2017).

1.3. Knowledge transfer in deep multitask learning

Hard parameter sharing is found on an initial common trunk of shared hidden layers from which task-specific heads branch out (Caruana, 1997; Long et al, 2017; Kendall et al., 2018; Liu and Shi, 2020). Although hard parameter sharing has proven to be beneficial when solving related learning tasks, without adequately inferring task relationships by propagating knowledge across task-specific branches such approaches are prone to *under transfer*, *i.e.*, to miss beneficial information (Long et al., 2017; Vandenhende et al., 2022). To additionally employ cross-task information, Dai et al. (2016) introduce Multitask Network Cascades, where the output of a task-specific branch is appended to the input of the next task-specific branch following a predefined causal cascade. Multi-linear Relationship Networks (Long et al., 2017) were proposed to learn task relationships on top of hard parameter sharing using tensor normal priors placed on the task-specific branches.

In contrast, soft parameter sharing involves its own set of hidden layers for each task in the encoder. Thereby, specifically designed modules handle information sharing adaptively to mitigate interferences among tasks (Misra et al., 2016; Yang and Hospedales, 2017). A constraint in soft parameter sharing approaches is scalability, as the size of the multitask network usually grows linearly with the number of tasks. Multitask Attention Networks (Liu et

al. 2019; Zheng et al., 2022) are based on a global encoder convolutional neural network (CNN). At various stages, features are passed to task-specific attention modules which adaptively learn their relevance for the respective task.

An additional strategy to address interferences in MTL is the rather indirect transfer of task-specific knowledge when adaptively balancing individual tasks as part of model optimization (*e.g.*, Kendall et al., 2018; Chen et al., 2018).

1.4. Exploitation of intermediate predictions

A few recent MTL approaches are built upon interim task predictions based on intermediate supervision (Lee et al., 2015; Gülçehre and Bengio, 2016) to leverage task interactions in the decoder stage of the model architecture (Vandenhende et al., 2022). In PAD-Net (Xu et al., 2018), a hard parameter sharing MTL network based on a CNN encoder is employed to derive initial estimations for multiple intermediate dense prediction tasks (*i.e.*, depth prediction, surface normal estimation, scene parsing, and contour detection). The resulting intermediate predictions are re-combined and subject to a spatial attention mechanism to obtain cross-task information for the final predictions. Vandenhende et al. (2020) expand the PAD-Net architecture to consider task interactions on multiple scales. Pattern-Affinitive Propagation Networks (PAP-Net; Zhang et al., 2019b) introduce the learning of pixel affinity matrices based on intermediate predictions to propagate cross-task affinitive patterns.

The methodological concept of utilizing initial predictions to improve generalization has also been employed in machine learning beyond deep learning in the past. *Stacked generalization* (Wolpert, 1992) is a meta-learning approach, that deploys the prediction outputs of models learned in the first stage to extend the feature space for learning a new model with improved generalization ability in the second stage. For classification problems, using class probabilities instead of the single predicted class outputs as input to the meta-learner turned out to be favorable in this course (Ting and Witten, 1999). In remote sensing, this multi-stage concept was recently leveraged for the post-classification enhancement of semantic image segmentation maps due to deep relearning (Zhu et al., 2021; Geiß et al., 2022a).

The principle of using the prediction outputs of preceding models to extend the feature space for subsequent models within multiple stages of learning was extended to multi-label classification (Godbole and Sarawagi, 2004; Read et al., 2011) and multi-target regression (Spyromitros-Xioufis et al., 2012). Meta-models are learned for each target variable with feature vectors augmented by prediction results for the residual target variables. This enables, knowledge sharing across the target variables and label dependencies can be exploited (Geiß et al., 2022b). Geiß et al. (2022b) exhaustively leverage this notion for information extraction from remote sensing imagery with a multi-target regressor chaining scheme.

Deep learning with intermediate prediction allows for the integration of multiple learning stages to exploit model outputs from preceding sub-models in a single end-to-end training realization. Capitalizing thereon for the *multitask classification* case, we consider the main contribution of this study as follows:

1) From a methodological point of view, we propose a deep MTL image classification framework that employs task-wise intermediately predicted class label probability distributions for the dynamic capturing of cross-task label interdependencies to result in models with enhanced accuracy properties. This is facilitated in two ways: *i)* interim class probability outputs are stacked to the final feature vector for classification (*i.e.*, *multitask stacking*); *ii)* interim class probability distribution outputs are employed to learn features explicitly internalizing label interdependencies using recurrent neural networks (RNNs). Following this path, interdependencies among the sequence of individual class probability values and interdependencies among the individual task-wise class label probability distributions are considered.
2) The proposed methodology is experimentally evaluated in the context of an innovative application domain. We address the multicriteria characterization of buildings based on SLI. In particular, this study approaches the accurate and efficient joint extraction of multiple natural hazards vulnerability-related building characteristics (*i.e., height*, *LLRS material*, *SBST*, *roof shape,* and *block position*). However, regardless of the application case of this study, the presented methods are applicable and potentially gainful under arbitrary image classification settings where multiple classification problems can be tackled simultaneously.
3) Within the addressed application and data setting, we carry out an exhaustive experimental evaluation of the presented multitask classification framework and its subcomponents including a systematic analysis of estimated generalization accuracies and associated training time consumptions.

The remainder of the paper is organized as follows. Section 2 provides a description of the data used and the proposed methodology. The experimental setup is pointed out in Section 3, while results are presented and discussed in Section 4. Section 5 concludes this paper.

2. Materials and methods

2.1. Street-level imagery

The employed SLI data comprises 204,030 GSV building façade views collected as part of Aravena Pelizari et al. (2021) within the 7 M inhabitant metropolitan area of Santiago de Chile, Chile: first, a spatially stratified sample of scenes with a perpendicular viewing direction w.r.t. the driving direction of the recording vehicle was acquired; next, the sampled imagery was subject to a CNN based filtering procedure to separate façade from non-façade views. Thereon, a reference dataset of 29,567 façade images has been labeled according to 3 multi-class target variables: *i)* the *material type of the LLRS* (MatLLRS), ii) building *height* (number of storeys) as well as *iii)* a *seismic building structural type* (SBST) characterizing a buildings' main-load bearing structure from the seismic vulnerability perspective (Geiß et al., 2015). To provide objectivity, an ontology jointly elaborated by local structural engineers and experienced image analysts was followed to specify the labels based on predefined visually inferable indicators (visual-structural criteria).

For this study, the reference data was extended by two additional attributes relevant to assess natural hazard risk and thus captured by designated building taxonomies (Silva et al., 2022):

roof shape (*RoofShp*) and *block position* (*BlockPos*). Fig. 2 provides an overview on all target variables, associated class labels as well as the criteria for assigning *RoofShp* and *BlockPos* (Fig. 2b; in case of ambiguity in labeling, *e.g.,* due to an unfavorable field of view or viewing angle, aerial imagery was considered). For in-depth details on façade image collection, target variables and labeling as depicted in Fig. 2a, we refer to Aravena Pelizari et al. (2021).

Height (6)	Material LLRS (7)	SBST (14)
1 storey (**H1**)	Unreinforced masonry (**MUR**)	**MUR_H1**
		MUR_H2-3
2 storeys (**H2**)	Confined masonry (**MCF**)	**MCF_H1-2**
		MR_H1-2
3-4 storeys (**H3-4**)	Reinforced masonry (**MR**)	**CR_H1-2**
		W,UNK_H1-2
	Reinforced concrete (**CR**)	**MCF_H3-4**
5-7 storeys (**H5-7**)		**MR_H3-4**
	Wooden and non-engineered (**W,UNK**)	**CR_H3-4**
		CR_H5-7
8-12 storeys (**H8-12**)		**CR_H8-12**
	Other commercial and industrial build. (**COM1,2,IND**)	**CR_H13+**
>13 storeys (**H13+**)		**COM1,2,IND_H1+**
	Other office build. (**COM3**)	**COM3_H8+**

a)

Roof shape (3)	
Flat (**FLT**)	• *flat (≤ ~7% slope)*
Pitched or gabled (**PIT_GAB**)	• *pitched or with gable ends* • *projecting dormers with intersecting pitched roofs*
Monopitch (**PIT_MON**)	• *unidirectional roofslope* • *lean-on roofs, where pitch rests against higher wall*

Block position (5)	
Detached single-party (**DET_SP**)	• *detached single-party with distance to neighboring buildings > ~4% lower building height*
Detached multi-party (**DET_MP**)	• *detached multi-party with distance to neighboring buildings > ~4% lower building height*
Semi-detached (**SDET**)	• *single-party* • *shared wall with one adjoining housing unit*
Adjoining block development (**ADJ_BD**)	• *part of dynamically grown block configuration*
Adjoining terraced (**ADJ_TR**)	• *terraced/row houses* • *adjoined to similar housing units* • *single-party*

b)

Fig. 2. Addressed tasks and multi-class manifestations (class numbers in brackets): a) Target variables as obtained by Aravena Pelizari et al. (2021)**;** b) the target variables assigned within this study. Respective labeling criteria are given in *italic* and were defined considering Allen et al. (2023).

Annotated façade image examples for all tasks and classes are presented in Fig. 3. Class-wise quantities of labeled images are visualized in Fig 7b.

Fig. 4 shows label co-occurrences within the compiled reference data as conditional probabilities to intuitively illustrate prevailing interdependencies across the considered classification tasks (Hua et al., 2019). *E.g.*, it is very likely that a building of *Height* H5-7 or higher has a reinforced concrete (CR) *MatLLRS*, a flat (FLT) *RoofShp*, and is a detached multi-party (DET_MP) building considering *BlockPos*. Besides, it can be observed that label co-occurrences are generally not symmetric. For instance, while $P(H1|MUR) = 0.18$, $P(MUR|H1) = 0.55$. It becomes apparent, that the relationships among various classes and tasks can manifest in complex but characteristic patterns. In this study, we model label interdependencies based on such patterns as encoded in intermediately predicted class-label probability distributions to improve prediction accuracy in multitask image classification.

2.2. Data balancing and data partition

The labeled reference data pool is subject to the label powerset (LP) transformation (*e.g.,* Charte et al., 2015). Thereby, each distinct combination of labels (labelset) is treated as a single label. In our MTL setting, the LP refers to the set of all occurring label combinations across the considered tasks. As such, the LP histogram permits an overall perspective on label co-occurrences and inherently provides evidence on label relationships. Thus, in order not to compromise the representativity of the reference data set nor the inherent information on label interdependencies, data balancing, and data partitioning consider the LP.

Fig. 3. Example façade imagery with class labels for the addressed vulnerability-related target variables Height, MatLLRS, SBST, RoofShp, and BlockPos.

2.2.1. LPRUS

Class imbalance in the reference data is attenuated by applying balancing at data level (Buda et al., 2018). Inspired by Charte et al. (2015), who address imbalance in multi-label classification, we propose a resampling strategy based on the LP histogram. Specifically, we implement *LP-based random undersampling* (LPRUS) of the most populated labelset bins, given the percentage of the data to keep as an input parameter (Fig. 7): first, the *cut-off point* at which accumulated frequencies meet the desired percentage value is determined, then, random samples are drawn accordingly (*i.e.,* sample size = *cut-off point*) from the identified bins (*i.e.*, labelset bins with a population > *cut-off point*). Residual instances are dropped.

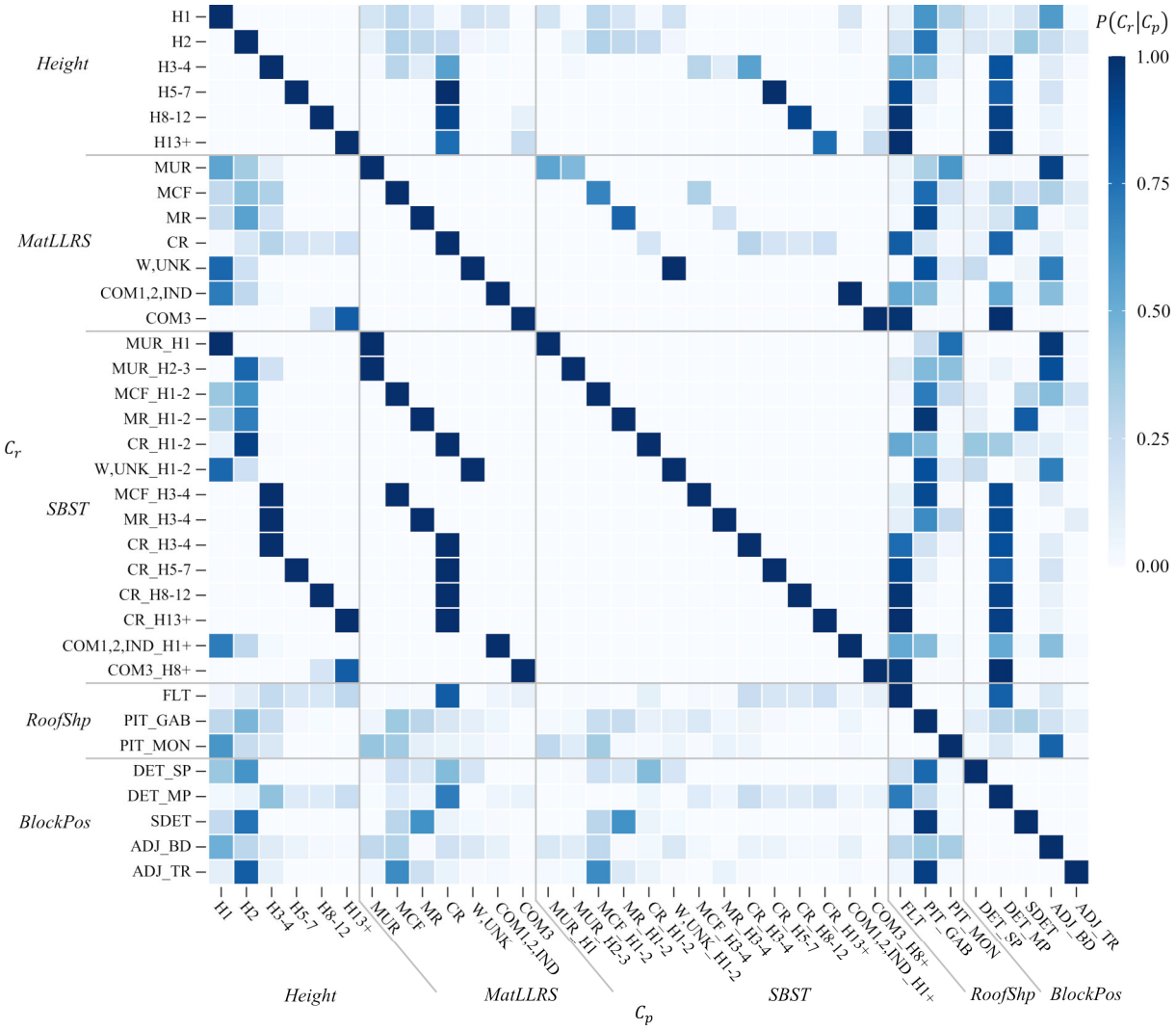


Fig. 4. Co-occurrence matrix of labels in the multitask annotated reference data set: y-axis labels denote the reference classes C_r, x-axis labels refer to potential co-occurrence classes C_p, the cells represent each class pair's conditional probability.

2.2.2. Data partition

To establish training, validation, and test data in consideration of occurring cross-task label combinations, we also build data partition upon the labelset bins. This is done by iteratively drawing random samples from the labelset bins until the desired data shares are reached. It is ensured that training, validation, and test set are spatially disjoint to avoid overoptimistic estimated model generalization capabilities (Geiß et al., 2017b). For this purpose, block-level spatial entity data (INE, 2018) are deployed.

2.3. Deep multitask learning with label interdependency modeling

The multitask classification (MTC) problem can be defined as follows: $X = \mathbb{R}^d$ denoting the d-dimensional instance space, each instance $x \in X$ is associated to a label space Y_T consisting of task specific class label sets $\{Y_{t_n}\}_{n=1}^{N}$; $t_n \in T$, $Y_{t_n} \in \{y_{t_n c_k}\}_{k=1}^{K}$.Thereby, N corresponds the cardinality of the total set of classification tasks $|T|$ and K to the respective task-specific numbers of classes. The goal of MTC is to learn a prediction model $M_T(x): X \rightarrow Y_T$.

We propose MTC with label interdependency modeling (MTC-LIM) to jointly accomplish multiple image classification tasks in an end-to-end trainable framework, that exploits cross-task label interdependencies. MTC-LIM (Fig. 5) is composed of four sub-modules: the *baseline MTC model* (BL), the *class-wise label interdependency model* (CLIM), the *task-wise label interdependency model* (TLIM), and a module for information *fusion and classification* (FC). Beyond the FC module, each sub-module is supervised via its individual loss respectively to guide learning toward reasonable label probability distributions.

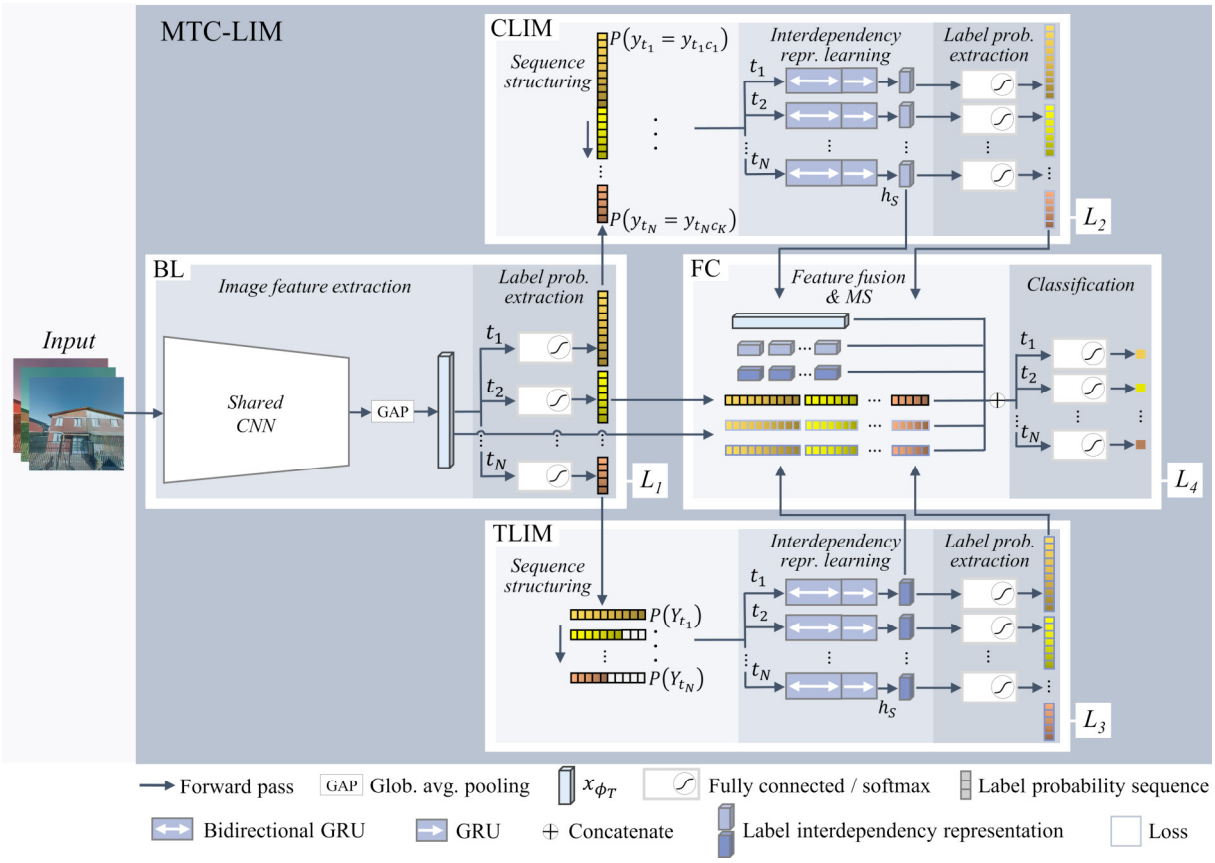


Fig. 5. The MTC-LIM framework is composed of 4 modules each supervised by its own loss function: in BL a hard parameter sharing CNN jointly learns spatial features and predicts task-wise interim class label probability distributions. Based thereon, CLIM and TLIM employ class-wise and task-wise label *interdependency representation learning* respectively. FC realizes the fusion of resulting representations, *multitask stacking,* and final prediction.

2.3.1. The BL module

The initial BL module serves for the joint learning of spatial features with regard to the considered tasks and thereon for the intermediate prediction of multitask class probability distributions. Specifically, hard parameter sharing MTL is deployed to simultaneously learn common image feature representations and a prediction model per task $M_{t_i}(x; \phi_T, \phi_{t_n}): X \rightarrow Y_{t_n}$. The parameters of the feature extractor ϕ_T are shared among T within a common encoder network $\{E; \phi_T\}$ and ϕ_{t_n} are the parameters of the task specific decoders $\{D_{t_n}; \phi_{t_n}\}$. Shared feature extraction is realized by an arbitrary CNN architecture and completed with *global average pooling* to keep the classification stages sparse. The output vector of the shared feature extraction stage can be denoted as $x_{\phi_T} = \phi_T(x)$.

All modules within the proposed MTC-LIM framework, including the BL module, employ fully connected layers with the *softmax* activation function, to transfer feature vectors to task-specific class-conditional probability distributions. Given a feature vector x_θ, the probability $P(y_{t_n} = c_k|x_\theta)$ of the kth category from the nth task is obtained by

$$P(y_{t_n} = c_k|x_\theta) = \frac{\exp(x_\theta w_{c_k} + b_{c_k})}{\sum_{k=1}^{K} \exp(x_\theta w_{c_k} + b_{c_k})}, \quad (1)$$

where w_{c_k} and b_{c_k} denote the weight vector and the bias of the kth neuron of the fully connected layer.

The task-specific class label probability distributions resulting from the BL module, *i.e.,*

$$P(Y_T|x_{BL}) = [P(Y_{t_1}|x_{BL}), P(Y_{t_2}|x_{BL}), \dots, P(Y_{t_N}|x_{BL})], \quad (2)$$

constitute the foundation for label interdependency modeling. In case of interdependency representation learning they are input to the CLIM and TLIM modules. When *multitask stacking* is applied they are also passed forward to the FC module.

2.3.2. Multitask stacking

Inspired by the works where multi-stage learning is used to exploit interdependencies across multiple target variables (Section 1.4), we propose *multitask stacking* (MS) to realize this concept in deep MTC. To leverage the outputs of multiple learning stages within a single end-to-end training realization MS uses intermediately supervised interim task predictions: MS concatenates the task-wise label probabilities predicted by the preceding modules with the corresponding feature vectors and uses this augmented feature vector for final classification (Fig. 5). In its basic case, MS is applied on top of hard parameter sharing MTL (BL module) in the FC module by

$$x_{BL-MS} = x_{BL} \oplus P(Y_T|x_{BL}), \quad (3)$$

$\oplus$ denoting vector concatenation. As such, MS enables the modeling of cross-task label interdependencies at the classification stage within the fully connected layers. Complementarily, when applying CLIM or TLIM, corresponding interim label probability distributions can be stacked to the resulting classification feature vector likewise. This can be interpreted as ensemble learning where each sub-model accounts for a different representation modality of the input (*i.e.*, *image features* and two types of label *interdependency representations*).

2.3.3. Label interdependency modeling with RNNs

The explicit modeling of interdependencies within the intermediately predicted class probability distributions is done with recurrent neural networks (RNNs; Chung et al., 2014), namely, *gated recurrent units* (GRUs; Cho et al, 2014). These receive the output class probability distributions from the BL module as properly arranged numerical sequences as input (Section 2.3.4.).

GRU is a modification of basic RNN proposed to adaptively capture dependencies in sequence data while avoiding defects of gradient vanishing and gradient explosion (Cho et al, 2014). They have shown powerful capabilities for modeling long-term dependencies (Chung et al., 2014; Jozefowicz et al., 2015; Mou et al., 2017). For this study, GRU was particularly chosen because of its sparsity, *e.g.*, compared to its alternative *long short-term memory units* (Hochreiter and Schmidhuber, 1997; Chung et al., 2014). Fewer parameters may be beneficial for the addressed application domain, which goes along with limited training data. In addition, it results in faster training and inference.

Generally, RNNs handle sequential input by having a recurrent hidden state whose activation at each time step is dependent on that of the previous time. Given a sequence data $z = (z_1, z_2, \dots, z_S)$, where $z_s, s \in \{1,2, \dots, S\}$ is the data at time step s an RNN updates its recurrent hidden state h_s by

$$h_s = \begin{cases} 0, & s = 0 \\ \varphi(h_{s-1}, z_s), & otherwise \end{cases}, \tag{4}$$

where φ denotes a nonlinear transformation function. As such, an RNN allows to model a probability distribution over the next element of the sequence data given its current state h_s being able to capture a time-dependent probability distribution over sequences of variable length (Chung et al., 2014).

A GRU (Fig. 6a) employs a transformation function, that incorporates two gating units: an update gate u_s and a reset gate r_s. The first controls how much the unit updates its activation or content. The latter controls how much the previously computed state is discarded. $\tilde{h}_s$ represents the current candidate activation. Thereon, the output activation of the hidden layer h_s is determined.

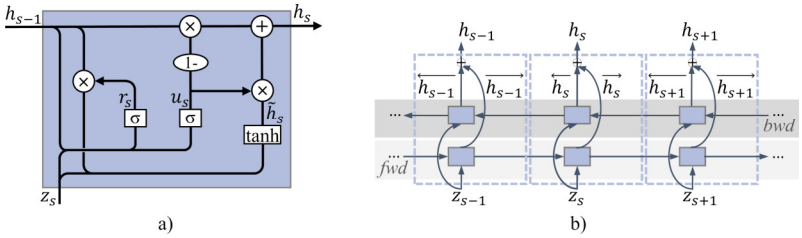


Fig. 6. a) GRU architecture and b) bidirectional GRU.

The recurrent transition of a GRU is calculated as follows:

$$u_s = \sigma(W_{zu} z_s + W_{hu} h_{s-1}), \tag{5}$$
$$r_s = \sigma(W_{zr} z_s + W_{hr} h_{s-1}), \tag{6}$$
$$\tilde{h}_s = \tanh\big(W_{zh} z_s + W_{hh}(r_s \times h_{s-1})\big), \tag{7}$$
$$h_s = (1 - u_s) \times h_{s-1} + u_s \times \tilde{h}_s, \tag{8}$$

where $\sigma(\cdot)$ denotes the logistic sigmoid function, $\tanh(\cdot)$ the hyperbolic tangent function, $\times$ element-wise multiplication, and W respective weight matrices.

As can be seen from Eq. 4-8, the recurrent connection allows for the dynamic modeling of contextual relationships along an input sequence. Ideally, the last hidden state h_S captures most

of the encoded information and thus can be employed to obtain the output for inference tasks such as multi-class classification (Mou et al., 2017; Hang et al., 2019).

In this study, we establish a *stacked* RNN composed of a *bidirectional* GRU followed by a *unidirectional* GRU (subsequently simply denoted as RNN) to comprehensively capture label interdependencies. The bidirectional GRU (Fig. 6b), consists of two identical GRUs that process the input data sequence in opposite direction and can be expressed as:

$$\overrightarrow{h_s} = GRU_{fwd}\left(z_s, \overrightarrow{h_{s-1}}\right), \tag{9}$$

$$\overleftarrow{h_s} = GRU_{bwd}\left(z_s, \overleftarrow{h_{s-1}}\right), \tag{10}$$

$$h_s = \overrightarrow{h_s} \oplus \overleftarrow{h_s}\,, \tag{11}$$

where $\overrightarrow{h_s}$ and $\overleftarrow{h_s}$ are the hidden states of the *forward* GRU and *backward* GRU respectively, and $\oplus$ denotes vector concatenation. This enables the exploitation of both, signals from previous information and signals from subsequent information (Schuster and Paliwal, 1997), which has already been shown beneficial for modeling class relationships in multilabel data (Hua et al., 2019).

2.3.4. The CLIM and TLIM modules

The label interdependency modeling modules in the first step perform a restructuring of the probability distributions obtained from the BL module (Fig. 5: *Sequence structuring*) to properly address their specific aim when being passed to the RNN in the following (Fig. 5: *Interdependency representation learning*).

CLIM aims at capturing the cross-task dependencies encoded among the sequence of all individual class label probabilities. Consequently, the class probability distributions for all tasks are concatenated to obtain the input sequence for the RNN:

$$z_{CLIM} = \left(P\left(y_{t_1 c_1}\right), \ldots, P\left(y_{t_1 c_K}\right), P\left(y_{t_2 c_1}\right), \ldots, P\left(y_{t_2 c_K}\right), \ldots, P\left(y_{t_N c_K}\right)\right). \tag{12}$$

As such, the RNN receives each item of the sequence as individual time step with its probability value as input feature. Correspondingly, the number of time steps is $S_{CLIM} = \sum_{n=1}^{N} K_{t_n}$.

TLIM in contrast is employed to extract inter-task dependencies among the task-wise class probability distributions. To this end, the BL output probability distributions are structured as an $N \times Kmax$ two-dimensional array:

$$z_{TLIM} = \begin{pmatrix} P\left(y_{t_1 c_1}\right), & \ldots, & P\left(y_{t_N c_1}\right) \\ \vdots & \ddots & \vdots \\ P\left(y_{t_1 c_{Kmax}}\right), & \ldots, & P\left(y_{t_N c_{Kmax}}\right) \end{pmatrix}, \tag{13}$$

where $Kmax$ refers to the highest number of class manifestations occurring among the considered classification tasks. Residuals between a considered task-specific class number K_{t_n} and $Kmax$ are zero-padded.

In this manner, it is enabled, that the RNN receives each task as an individual time-step with its class probability distribution as input features.

z_{CLIM} and z_{TLIM} are input to a proper RNN per task to model label interdependencies task-specifically. The hidden layer states of the last time step are respectively considered as label *interdependency representations* and passed forward to fully connected layers with *softmax* activation. This facilitates obtaining label interdependency-based interim prediction outputs and thereon intermediate supervision. Resulting label interdependency representations and interim predictions are input to the FC module. The fact that label interdependency is modeled at the level of estimated output probability sequences rather than at the image feature level keeps the number of additional trainable parameters due to CLIM and TLIM relatively low (Table 1).

2.3.5. The FC module

The FC module is responsible for feature fusion, MS, and final prediction. Herein, the features and interim class label probability distribution outputs from the preceding modules can be concatenated to a final feature vector. This feature vector is passed to a fully connected layer with *softmax* activation for each task to lastly obtain the classification output. In this manner, the FC module aims at exploiting the resulting multi-view perspective on the image data (Aravena Pelizari et al., 2018) considering the complementary feature sub-spaces *image features*, label *probability distributions* (MS) as well as inter-class and inter-task label *interdependency representations*.

2.3.6. Optimization and inference

During training, given a labeled example $(x, \{y_{t_n}\})$, the MTC model learns by updating the shared and task-specific parameters to jointly minimize *categorical cross entropy* for each task. Thereby, MTC loss is defined as the sum of all task-specific losses:

$$L_m = \sum_{n=1}^{N} L_{t_n}. \quad (14)$$

Thereon, the overall optimization objective is calculated as the sum of all considered sub-model specific losses, *e.g.*, $L_{MTC-LIM} = \sum_{m=1}^{4} L_m$.

During inference the categorical labels are obtained from the probabilistic outputs by $argmax\, P(y_{t_n} = y_{t_n c_k})$.

3. Experimental setup

The percentage of the data to be kept in the LPRUS balancing of the reference data is set to 85%. Training, validation, and test data shares in the labelset-based data partition are set to 65%, 17.5%, and 17.5%, respectively.

To integrate different CNN architectures and design concepts (*architecture engineering* vs. *neural architecture search*), we implement *DenseNet121* (Huang et. al., 2017) and *EfficientNetV2-B3* (Tan and Le, 2021) as feature extractors within the BL module. Both

networks focus on parameter efficiency, *i.e.*, a favorable trade-off between parameter sparsity and accuracy properties (ibid.; Table 1).

Table 1. Deployed frontend CNN-Architectures, input image sizes, *ImageNet* top-1 accuracies, and trainable parameternumbers when addressing all 5 tasks

Feature extractor	Input size [Pixels]	ImageNet acc. [%]	# param. to train (5 tasks) [M]		
			STC	MTC-BL	MTC-LIM
DenseNet121	224^2	75.00	34.81	6.99	7.14
EfficientNetV2-B3	300^2	82.00	64.16	12.88	13.04

For training, frontend CNNs are initialized with *ImageNet* (Russakovsky et al., 2015) pretrained parametrization. Fully connected layers are initialized with *He normal* initialization (He et al., 2015) and subject to *L2* kernel weights regularization ($L2 = 0.0001$). The GRUs are initialized with a *Glorot uniform* initializer (Glorot and Bengio, 2010). The number of neurons is set identical for both, the bidirectional and the unidirectional GRU, to 32 in the CLIM module and 12 in the TLIM module. All models are trained uniformly using Adaptive Moment Estimation optimization (Kingma and Ba, 2014) with an initial learning rate of 0.0001. For exhaustive but efficient training, the learning rate is reduced by a factor of 0.5 at validation accuracy plateaus. Early stopping is applied, to mitigate overfitting. Considering the computational resources of the Nvidia Quadro RTX 4000 GPU with 8 GB memory all networks are trained with a batch size of 24.

The performed experiments are intended to provide reliable evidence on the accuracy properties of the different feature vector configurations available depending on the modules connected. Estimated generalization capabilities are reported as overall accuracy (OA), Cohen's kappa statistic (κ), and class-wise F_1-scores obtained from 11 independent realizations.

Additionally, we provide the MTC performance of method m as the accumulated residuals in accuracy per task w.r.t. its single task classification (STC) counterparts b (*i.e.,* the STC models with the same feature extractor):

$$\Delta_m = \sum_{n=1}^{N}(M_{m,t_n} - M_{b,t_n})/M_{b,t_n}, \tag{15}$$

M_{t_n} referring to the performance measure of task t_n.

Starting from the base benchmark STC, we employ hard parameter sharing MTC (BL) followed by *multitask stacking* (MS), label *interdependency representation learning* (CLIM and TLIM), and their combinations. Table 2 provides an overview on all experiments listing employed modules, feature vectors for classification, and supervised losses respectively.

Table 2. Conducted experiments, involved modules, feature vector composition, and supervision (see Fig.5). STC refers to single task classification. Classification feature vectors for MTC are indicated by the learned representations (x_{module}) and interim cross-task class probability distributions $P(Y_T|x_{module})$ used.

Experiment	Modules	Classification feature vector	Supervision
STC	-	$[x_{STC}]$	L_{STC}
BL	BL	$[x_{BL}]$	L_1
BL-MS	BL, FC	$[x_{BL}, P(Y_T\|x_{BL})]$	L_1, L_4
CLIM	BL, CLIM, FC	$[x_{BL}, x_{CLIM}]$	L_1, L_2, L_4
CLIM-MS$_{BL}$	BL, CLIM, FC	$[x_{BL}, x_{CLIM}, P(Y_T\|x_{BL})]$	L_1, L_2, L_4
CLIM-MS$_{ALL}$	BL, CLIM, FC	$[x_{BL}, x_{CLIM}, P(Y_T\|x_{BL}), P(Y_T\|x_{CLIM})]$	L_1, L_2, L_4
TLIM	BL, TLIM, FC	$[x_{BL}, x_{TLIM}]$	L_1, L_3, L_4
TLIM-MS$_{BL}$	BL, TLIM, FC	$[x_{BL}, x_{TLIM}, P(Y_T\|x_{BL})]$	L_1, L_3, L_4
TLIM-MS$_{ALL}$	BL, TLIM, FC	$[x_{BL}, x_{TLIM}, P(Y_T\|x_{BL}), P(Y_T\|x_{TLIM})]$	L_1, L_3, L_4
LIM	BL, CLIM, TLIM, FC	$[x_{BL}, x_{CLIM}, x_{TLIM}]$	L_1, L_2, L_3, L_4
LIM-MS$_{BL}$	BL, CLIM, TLIM, FC	$[x_{BL}, x_{CLIM}, x_{TLIM}, P(Y_T\|x_{BL})]$	L_1, L_2, L_3, L_4
LIM-MS$_{ALL}$	BL, CLIM, TLIM, FC	$[x_{BL}, x_{CLIM}, x_{TLIM}, P(Y_T\|x_{BL}), P(Y_T\|x_{CLIM}), P(Y_T\|x_{TLIM})]$	L_1, L_2, L_3, L_4

4. Results and discussion

4.1. Data balancing and partition

As first step, the multitask labeled dataset was subject to the LPRUS undersampling strategy (Section 2.2.1) to mitigate class imbalance without losing information on label interdependencies. Fig. 7a shows the top 25 labelset bins as well as the identified *cut-off point*. The resulting amounts of samples to be kept and dropped are indicated in orange and blue. Fig. 7b highlights the effect of LPRUS on the task-wise class histograms. It can be observed, that LPRUS leads to an undersampling of the majority bins in particular and therefore allows for attenuating class imbalance in multitask annotated data.

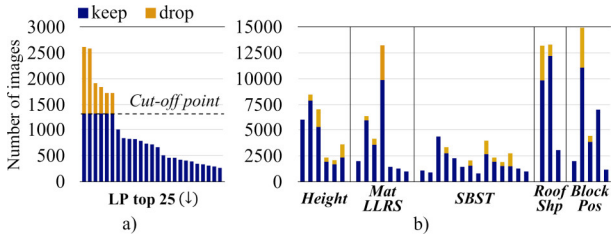


Fig. 7. LPRUS. a) Top 25 labelset bins with highest frequencies arranged in descending order, determined *cut-off point,* kept samples, and dropped instances. b) Effect on task-wise class histograms.

The outcome of LPRUS was split up into training, test, and validation data sets (Fig. 8) containing 16,191, 4,469, and 4,473 images respectively.

4.2. Comparative model accuracies

Insights on the impact of the different employed methodological components, *i.e.*, the parameter sharing within the BL module, MS, the learning of label interdependency representations with CLIM and TLIM as well as their combinations are revealed by evaluating 12 different model configurations (Table 2) for the two frontend CNN architectures each. The mean test set accuracies of the employed model configurations in terms of OA, κ, and Δ_m are provided in Table 3.

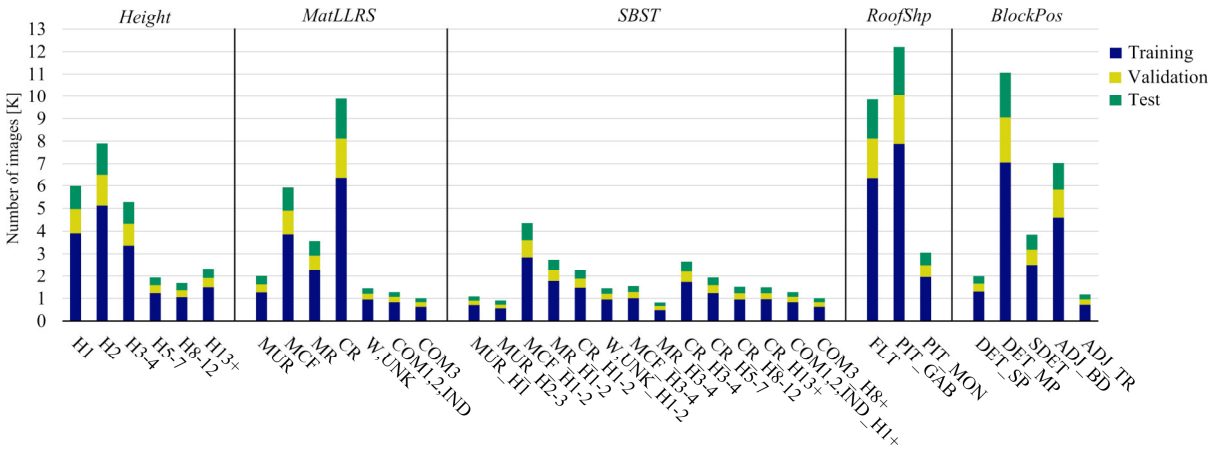


Fig. 8. Class frequency distributions across training, test, and validation data for the 5 considered tasks.

In addition, to obtain a more comprehensive view on the comparative generalization capabilities of the employed models, we depict all results as boxplots of the accumulated residuals in accuracy w.r.t. the STC run with the *median* of cross-task mean κ statistics ($\Delta_m(median)$; Fig. 9). Each boxplot highlights median values, interquartile ranges, whiskers, and potential outliers. Concerning STC, all possible combinations of the individual task-wise outputs are considered.

In consideration of the results for all tasks, it can be generally stated that with the presented data setup and methods viable accuracy levels beyond a mean OA = 79.90% and a mean κ = 77.94% (Table 3, STC with *DenseNet121* frontend on *SBST* task) can be achieved. Thereby, estimated classification accuracies of the models with *EfficientNetV2-B3* frontend are higher than their *DenseNet121* counterparts. This is in line with their accuracy scores for the ImageNet data set (Table 1), although the superiority of *EfficientNetV2-B3* is not similarly pronounced here. Regardless of the frontend CNN architecture it can be seen, that hard parameter sharing alone (BL) already increases accuracy compared to the STC results. The proposed MS allows for further gains in accuracy (BL-MS). Subsequently, models containing label *interdependency representation learning* via the CLIM module, the TLIM module, or both modules outperform BL-MS predictions. This reflects their ability to better capture beneficial information encoded within the intermediately predicted label probability sequences. Correspondingly, the employed RNNs are revealed as appropriate to dynamically account for label interdependencies within the proposed framework.

Looking at the results obtained with the *DenseNet121* frontend (Table 3; Fig 9a), MS on top of label *interdependency representation learning* with either CLIM or TLIM (as in CLIM-MS_{BL} and CLIM-MS_{ALL} as well as TLIM-MS_{BL} and TLIM-MS_{ALL}) does not further increase accuracy. The results of the LIM models, which combine CLIM and TLIM, in contrast, show consecutive accuracy gains when MS is employed (LIM-MS_{BL} and LIM-MS_{ALL}). With a mean OA = 87.13%, a mean κ = 82.79% as well as corresponding $\Delta_m(mean)$ values of +5.34% and +7.27% respectively, the LIM-MS_{ALL} configuration performs best when building upon the *DenseNet121* frontend.

The results with the *EfficientNetV2-B3* frontend (Table 3; Fig 9b), show a different accuracy pattern at this point. Here, MS on top of label *interdependency representation learning* by either CLIM or TLIM allows for a further increase in accuracy. This accounts for the model

configurations CLIM-MS$_{BL}$ as well as TLIM-MS$_{BL}$ and TLIM-MS$_{ALL}$. For the latter two, this becomes obvious when considering the corresponding boxplots. Augmenting the initial feature space with CLIM label interdependency representations and BL probability MS (CLIM-MS$_{BL}$) is revealed as the most accurate of all model configurations, resulting in a mean OA = 88.03%, a mean κ = 83.69% as well as respective $\Delta_m(mean)$ values of +4.80% and +6.37%.

Table 3. Mean accuracy values [%] obtained from 11 independent trials.

	Runs	*Height*		*MatLLRS*		*SBST*		*RoofShp*		*BlockPos*		*Mean*		$\Delta_m(mean)$	
		OA	κ	OA	κ	OA	κ	OA	κ	OA	κ	OA	κ	OA	κ
DenseNet121	STC	89.31	86.37	86.12	81.60	79.90	77.94	88.71	81.01	87.10	81.17	86.23	81.62	-	-
	BL	89.14	86.15	86.74	82.41	80.64	78.76	88.70	81.02	87.45	81.69	86.54	82.01	+1.86	+2.44
	BL-MS	89.67	86.82	86.93	82.66	81.21	79.39	88.62	80.86	87.54	81.84	86.80	82.31	+3.40	+4.31
	CLIM	89.56	86.68	87.21	83.04	81.43	79.63	88.95	81.43	87.56	81.87	86.94	82.53	+4.25	+5.67
	CLIM-MS$_{BL}$	89.56	86.69	87.04	82.82	81.36	79.55	88.94	81.42	87.39	81.64	86.86	82.42	+3.78	+5.00
	CLIM-MS$_{ALL}$	89.45	86.54	87.07	82.85	81.14	79.32	88.94	81.41	87.40	81.63	86.80	82.35	+3.41	+4.56
	TLIM	**89.72**	**86.89**	87.19	83.01	81.48	79.68	88.88	81.34	87.81	82.23	87.02	82.63	+4.70	+6.27
	TLIM-MS$_{BL}$	89.68	86.84	87.25	83.10	81.47	79.68	88.82	81.23	87.60	81.93	86.97	82.56	+4.40	+5.82
	TLIM-MS$_{ALL}$	89.40	86.48	87.12	82.92	81.19	79.36	88.93	81.41	87.63	81.99	86.85	82.43	+3.74	+5.07
	LIM	89.63	86.78	87.34	83.23	81.47	79.68	88.89	81.34	87.61	81.96	86.99	82.60	+4.53	+6.08
	LIM-MS$_{BL}$	89.60	86.74	87.40	83.31	81.49	79.71	88.98	81.49	**87.87**	**82.33**	87.07	82.71	+5.01	+6.80
	LIM-MS$_{ALL}$	89.70	86.87	**87.46**	**83.38**	**81.61**	**79.83**	**89.14**	**81.76**	87.73	82.12	**87.13**	**82.79**	**+5.34**	**+7.27**
EfficientNetV2-B3	STC	89.74	86.92	87.80	83.83	81.58	79.80	89.19	81.90	87.73	82.13	87.21	82.92	-	-
	BL	89.09	86.10	88.80	85.16	82.15	80.42	89.27	82.00	87.72	82.17	87.40	83.17	+1.19	+1.59
	BL-MS	89.46	86.58	88.80	85.18	82.43	80.74	89.45	82.33	87.99	82.55	87.63	83.47	+2.46	+3.43
	CLIM	89.75	86.94	88.97	85.37	82.77	81.10	**89.52**	**82.45**	88.12	82.75	87.83	83.72	+3.62	+4.92
	CLIM-MS$_{BL}$	**90.15**	**87.45**	89.13	85.59	**83.16**	**81.53**	89.51	82.42	**88.19**	**82.84**	**88.03**	**83.96**	**+4.80**	**+6.37**
	CLIM-MS$_{ALL}$	89.77	86.96	88.92	85.31	82.83	81.16	89.43	82.31	88.12	82.74	87.81	83.70	+3.56	+4.77
	TLIM	89.44	86.54	88.91	85.33	82.69	81.02	89.44	82.32	88.11	82.72	87.72	83.59	+3.01	+4.11
	TLIM-MS$_{BL}$	89.58	86.73	88.92	85.32	82.67	80.98	89.38	82.23	87.93	82.47	87.70	83.55	+2.88	+3.86
	TLIM-MS$_{ALL}$	89.41	86.51	89.01	85.44	82.56	80.87	89.45	82.33	87.88	82.39	87.66	83.51	+2.68	+3.64
	LIM	89.63	86.79	88.96	85.38	82.74	81.06	89.49	82.39	87.97	82.54	87.76	83.63	+3.24	+4.40
	LIM-MS$_{BL}$	89.80	87.01	**89.17**	**85.65**	82.92	81.27	89.43	82.29	88.09	82.72	87.88	83.79	+3.96	+5.32
	LIM-MS$_{ALL}$	89.63	86.79	89.05	85.51	82.87	81.22	89.47	82.35	88.02	82.61	87.81	83.69	+3.53	+4.76

0 *max*

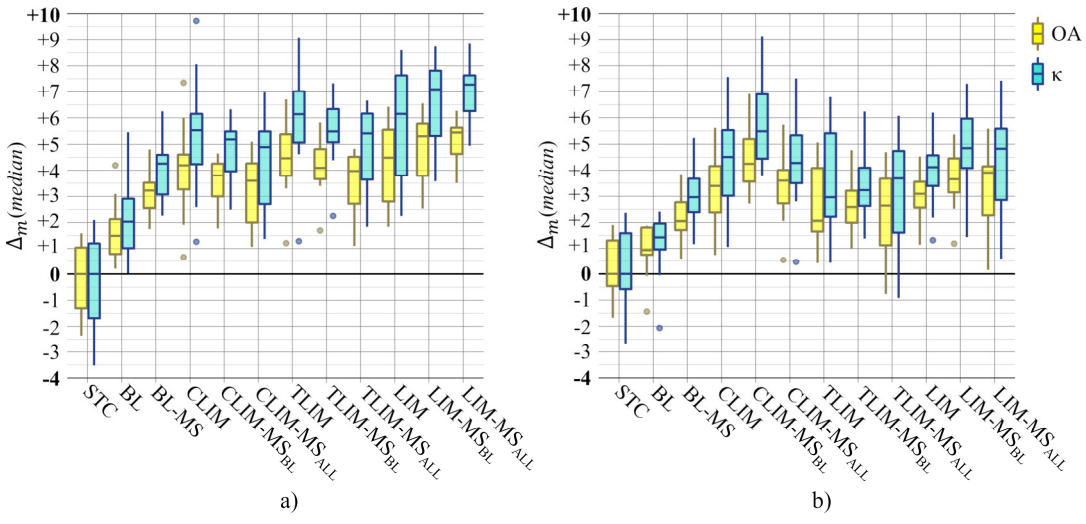


Fig. 9. Comparative accuracies of multitask classifications as accumulated residuals w.r.t. the STC median [%]: a) with *Densenet121* frontend, b) with *EfficientNetV2-B3* frontend.

Considering the task-wise accuracy means, it can be stated that the proposed methods allow for obtaining accuracy gains across all individual tasks.

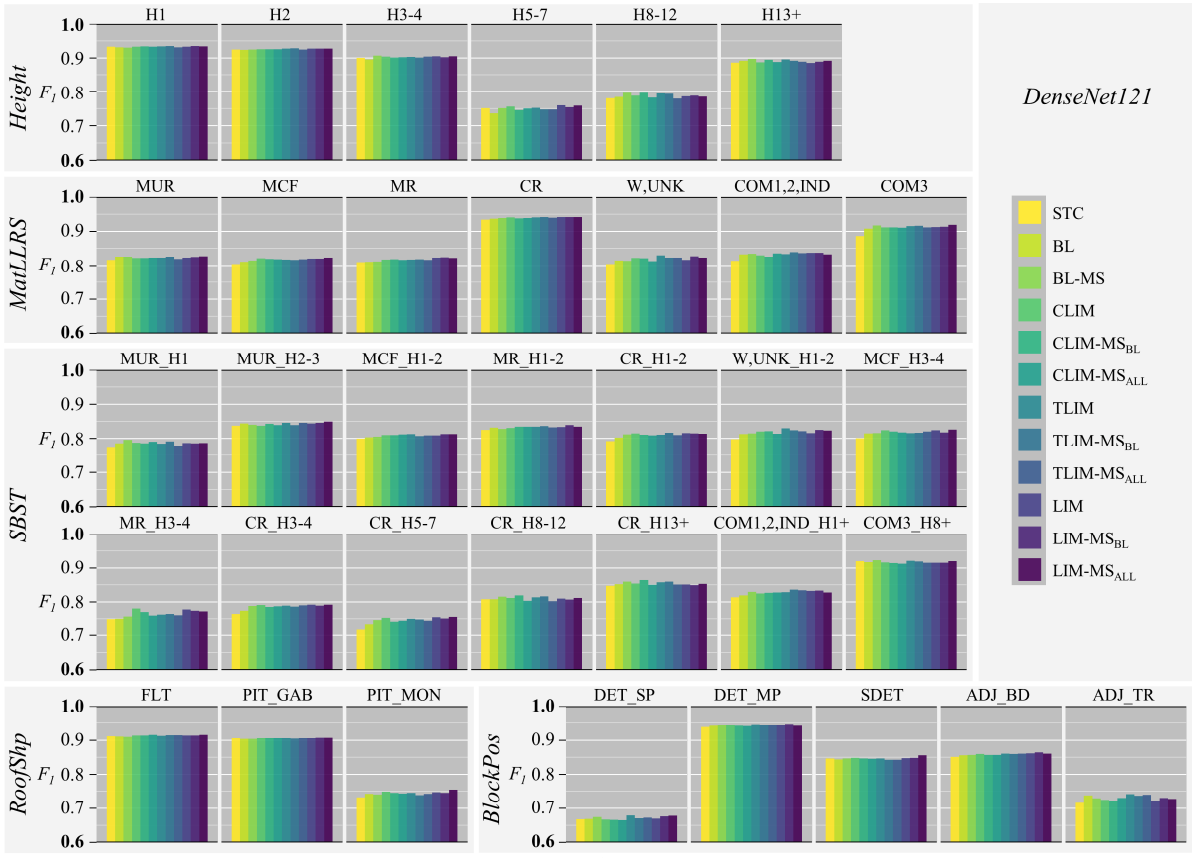
The boxplots reveal, that due to the stochastic processes occurring in the course of model training accuracy value ranges can spread considerably. Nonetheless, median values, interquartile ranges, and whiskers, mostly follow the accuracy hierarchy indicated by the mean accuracy values (Table 3). It is confirmed that with both frontends *DenseNet121* (Fig. 9a) and *EfficientNetV2-B3* (Fig. 9b), a combination of *interdependency representation learning* and MS performs best in terms of estimated generalization accuracies, *i.e.*, LIM-MS$_{ALL}$ with median values of +5.44% OA and +7.27% κ and CLIM-MS$_{BL}$ with median values of +4.25% OA and +5.49% κ. A closer look at the boxplots also indicates that individual model configurations allow for higher accuracy gains than reflected by mean and median values. *E.g.*, the difference between the *DenseNet121* TLIM upper whisker and the corresponding STC whisker is +5.12% OA and +6.98% κ. The difference between the upper whisker of *EfficientNetV2-B3* CLIM-MS$_{BL}$ and the corresponding STC whisker is +5.01% OA and +6.76% κ. Analogously, CLIM-MS$_{BL}$ surpasses BL by +5.04% OA and +6.71% κ. Overall, CLIM-MS$_{BL}$ with *EfficientNetV2-B3* frontend achieves the best estimated generalization ability among all performed realizations resulting in a cross-task mean accuracy value of 88.43% OA, and 84.49% κ (confusion matrices are show in Fig. 11, task- and class-wise accuracy values in Fig. 13).

Concerning the presented methods, we can conclude at this point: *i)* the inclusion of MS (BL-MS) outperforms hard parameter sharing MTL (BL) alone, *ii)* label *interdependency representation learning* via CLIM and/or TLIM (CLIM, TLIM and LIM) outperforms MS (BL-MS), indicating that it allows for an enhanced capturing of cross-task label interdependencies *iii)* the combination of MS and label *interdependency representation learning* allows for synergies and results in the highest mean and median accuracy values, *e.g.*, *DenseNet121* LIM-MS$_{ALL}$ and *EfficientNetV2-B3* CLIM-MS$_{BL}$.

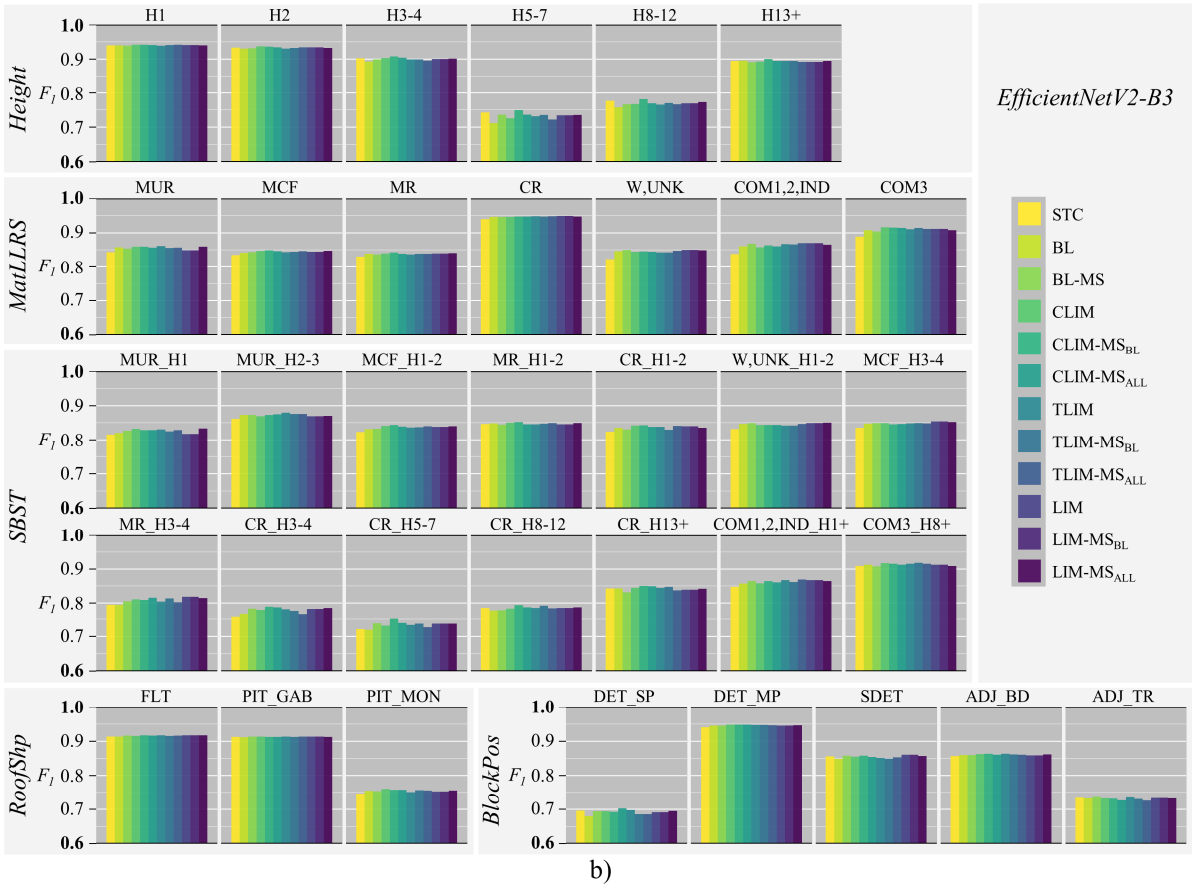
Fig. 10 shows the class-wise mean F_1-scores of the obtained results. In agreement with the presented aggregated evaluations, also on class-level, the highest accuracy values are obtained by MTL model configurations considering label interactions due to MS, label *interdependency representation learning*, or, as in most of the cases, a combination of both.

For multiple classes deteriorations can be observed when comparing the accuracies of BL MTL with STC (*negative transfer*), these include H1, H2, H3-4, H5-7, CR_H8-12, COM3_H8+, FLT, PIT_GAB and SDET for the *DenseNet121* frontend runs (Fig. 10a) as well as H1, H2, H3-4, H5-7, H8-12, MR_H3-4, CR_H5-7, CR_H8-12, PIT_GAB, DET_SP, SDET and ADJ_TR for the *EfficientNetV2-B3* frontend runs (Fig. 10b). The barplots indicate, that interdependency modeling counteracts such deteriorations.

The figures further indicate, that particularly the MTC prediction accuracies of classes less represented in the training data (Fig. 8) can be improved by explicitly accounting for label interdependencies, *e.g.*: H5-7, H8-12, H13+ for *Height*; MUR, MR, W,UNK, COM1,2,IND, COM3 for *MatLLRS*; MUR_H1, MUR_H2-3, CR_1-2, MR_H3-4, CR_H3-4, CR_H5-7, CR_H8-12, CR13+, COM1,2,IND_H1+ for *SBST*; PIT_MON for *RoofShp* and DET_SP for *BlockPos*.



a)



b)

Fig. 10. Class-wise mean F_1-scores for the addressed classification tasks and employed model configurations. a) With Densenet121 frontend, b) with EfficientNetV2-B3 frontend.

4.3. Insights on class-wise accuracy levels

To provide background insights on the general accuracy levels of the individual classes as well as associated prediction errors Fig. 11 depicts the confusion matrices resulting from the best performing MTC model.

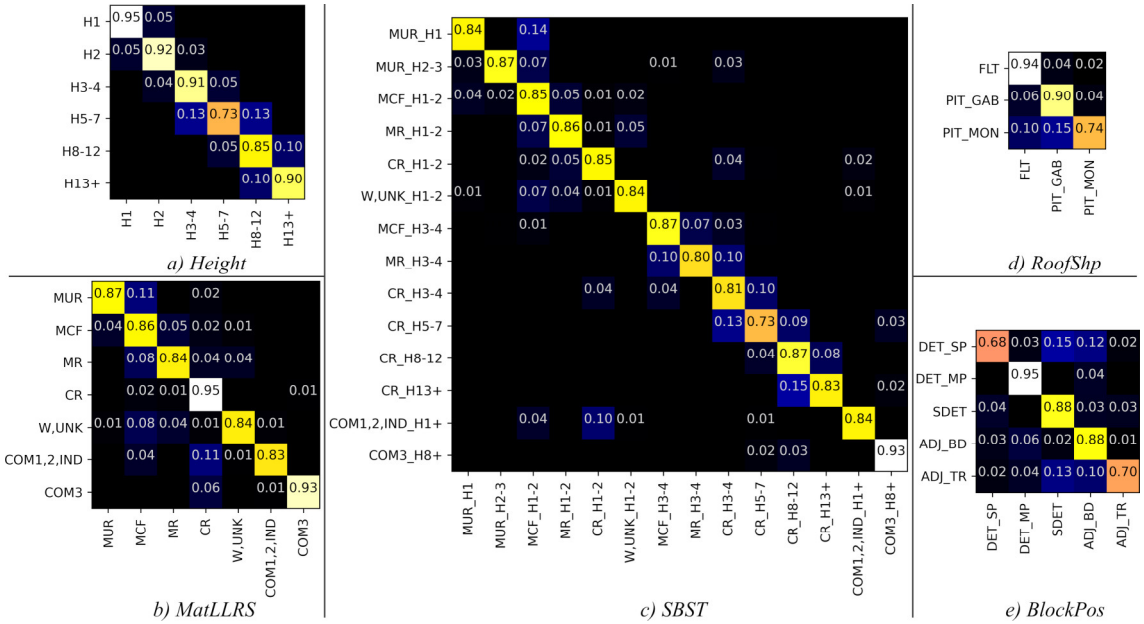


Fig. 11. Confusion matrices of best performing MTC model for the five tasks (y-axis: reference labels, x-axis: predicted labels).

For the *Height* predictions estimated commission and omission errors occur exclusively among adjoining classes (*i.e.*, classes one height class above or below; Fig. 11a). H5-7 is revealed to be particularly error prone with comparatively high errors of omission. *MatLLRS* classifications, in contrast, exhibit errors across several classes (Fig. 11b). This applies, *e.g.*, for the widespread MCF which has been built for a relatively long time and thus comprises a large within class variance. When buildings are plastered or painted, the visual characteristics to differentiate MCF from other LLRS materials can be subtle (Aravena Pelizari et. al, 2021). Moreover, there are errors of omissions for COM1,2,IND due to misclassification as CR. This is reasonable, COM1,2,IND is a residual class for commercial and industrial buildings not unambiguously assignable to the other *MatLLRS* classes or whose occurrence is too low to be adequately represented in an appropriate additional LLRS material category. Such buildings can feature high commonalities in their visual characteristics with CR buildings, which could hinder their distinction. The error patterns of *Height* and *MatLLRS* classification are reflected by the SBST predictions: errors predominantly occur among buildings with same height but different LLRS material or among adjoining height manifestations (Fig. 11c). Here also the *SBST* of height H5-7 (CR_H5-7) shows the highest estimated error rates. Regarding *RoofShp* prediction, PIT_MON is challenging (Fig. 11d). PIT_MON roofs often refer to MUR or MCF buildings where the roof slopes backward away from the street (Fig. 1a). This is not directly visible in a frontal façade view and needs to be inferred indirectly from other façade characteristics. Concerning *BlockPos*, DET_SP, and ADJ_TR are especially affected by errors of omission, *i.e.*, images wrongly predicted as SDET and ADJ_TR (Fig. 11e).

The identified error prone classes are in many cases also minority classes in the training data. Looking back at the comparative accuracies (Fig. 10) it can be observed that these more challenging classes can particularly benefit from the modeling of label interdependencies (*e.g.*, *Height* class H5-7, *MatLLRS* class COM1,2,IND, SBST class CR_H5-7, *RoofShp* PIT_MON, and *BlockPos* DET_SP).

4.4. Consideration of training times

A comparative overview on the mean training time for the employed model configurations is given in Fig. 12.

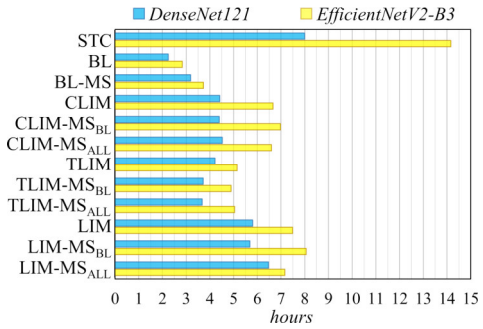


Fig. 12. Mean training time per model configuration required to address all tasks.

It can be noticed, that for the addressed data setting and application scenario the training time consumption of STC can be substantially reduced by using MTL models while the estimated generalization accuracies simultaneously improve (Table 3). Looking at the models with *DenseNet121* frontend the configuration with the highest accuracy values (LIM-MS$_{ALL}$) allows for mean training time reduction of 1.52h (19.02%) with a $\Delta_m(mean) =$ +5.34% OA and $\Delta_m(mean) =$ +7.27% κ. The remaining *DenseNet121* frontend model configurations allow for a more pronounced reduction in training time, however, with lower accuracy gains. *E.g.*, TLIM reduces mean training time compared to STC by 3.78h (47.31%) with only minor accuracy degradations attaining a $\Delta_m(mean) =$ +4.70% OA and $\Delta_m(mean) =$ +6.27% κ.

For the models with *EfficientNetV2-B3* front end, the gain in terms of accuracy/training time-efficiency is even more distinct. Here, the model configuration with the highest estimated accuracy values (CLIM-MS$_{BL}$) already results in a mean training time reduction of 7.18h (50.71%), while obtaining $\Delta_m(mean)$ values of +5.01% OA and +6.76% κ.

The presented experimental evaluations demonstrate that the presented methods allow for considerable gains in generalization capability estimates compared to STC but also compared to hard parameter sharing MTL (BL) alone. The highest accuracy values are achieved by combing MS and label *interdependency representation learning*. Simultaneously, training time consumption is substantially reduced compared to STL.

4.5. Application: Spatially distributed building characteristics

To illustrate the application of the proposed MTL framework, *i.e.*, large-area building characterization for multihazard risk assessments (Section 1.1), we employ the best performing MTC-model (*EfficientNetV2-B3* CLIM-MS_{BL} configuration) to classify all 204,030 façade views. Fig. 13 depicts the spatial distributions of each of the predicted classes for the five addressed tasks. Furthermore, resulting class frequency distributions as well as corresponding class-wise F_1-scores, OA, and κ values are presented. The maps reveal the distinct patterns for each class of the addressed building characteristics across the Santiago de Chile area. The individual patterns are the result of the urban growth history, past earthquakes, the associated evolvement of building codes as well as the impact of socioeconomic factors (*e.g.*, demography, income, economy) on construction (Aravena Pelizari et. al, 2021).

In synopsis, the maps to a certain degree also reflect the spatial manifestations of co-occurrences (*i.e.*, interdependencies) among the addressed classes (see also Fig. 4). For instance, buildings assigned to the *Height* class 5-7 storeys (H5-7) or a higher class, almost all refer to detached multiparty (DET_MP), reinforced concrete (CR), or office (COM3) constructions with a flat roof (FLT). Such buildings predominantly occur in the residential areas of medium to high socioeconomic status and the business and financial districts, which particularly expand northeastwards from the center. The locations of COM3 buildings are exclusively limited to such sectors (for spatial consideration of Greater Santiago from a socio-economic perspective we refer to Garreton, 2017). Reinforced masonry (MR) as well as wooden and non-engineered (W_UNK) constructions in contrast mainly cover low to medium socioeconomic status residential buildings and extend radially from the city center and the high economic status sectors in the northeast. Such buildings predominantly cover constructions of 1-2 storeys height (H1, H2) with pitched or gabled (PIT_GAB) roofs. Thereof, the MR buildings mainly refer to semi-detached (SDET) buildings, while the dominant *BlockPos* class for W_UNK buildings is adjoining block development (ADJ_BD) followed by detached single-party (DET_SP). As a further example, unreinforced masonry (MUR) buildings particularly concentrate within municipalities in the core of Santiago and its direct surroundings. These sectors cover the historic extent, from which the city began to expand. Thereby, the higher MUR buildings with 2-3 storeys (MUR_H2-3) are primarily situated in the historic city center. The core of Santiago de Chile has been highly densified and MUR buildings most frequently belong to adjoined block developments. With regard to their roof shapes, MUR buildings most frequently comprise monopitch (PIT_MON) followed by PIT_GAB roofs.

The outlined examples are intended to once more illustrate occurring interdependencies among the class manifestations of the addressed building characteristics. From an application point of view, it becomes apparent that their quantities such as their spatial distributions are highly heterogeneous and each follows a very distinct spatial pattern. Accordingly, also the vulnerability of the classified buildings to natural hazards comprises distinct spatial variabilities, making such information highly valuable for the spatial modeling of risk (Gomez-Zapata et. al, 2021; Geiß et. al, 2022c). Such details are a decisive component when it comes to tailored measures in the context of pre- and post-event emergency management.

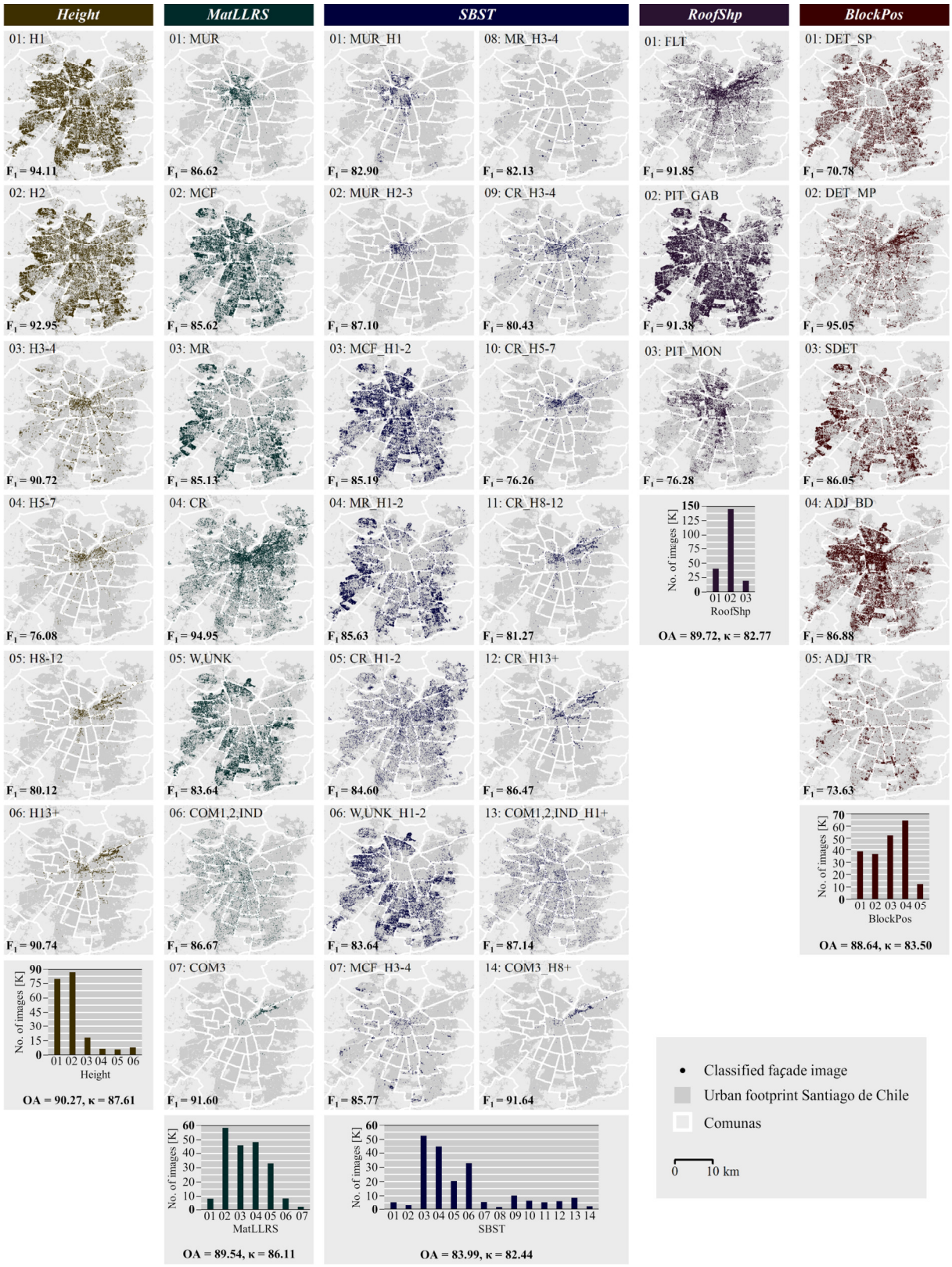


Fig. 13. Spatial distribution, class frequencies, and accuracy measures [%] of classification output from best performing MTC model. Built-up area as given by the Global Urban Footprint (Esch et al., 2017) and the administrative boundaries of the covered municipalities (comunas) are used as backdrop.

5. Summary and conclusion

This paper introduced a methodological framework to dynamically leverage cross-task label interdependencies in end-to-end deep MTL image classification. Interdependencies are modeled based on task-specific interim class label probability distribution estimates by a classical hard parameter sharing CNN frontend in two ways: *i)* interim class probability outputs are stacked to the final feature vector for classification (*i.e.*, *multitask stacking*); *ii)* interim class probability distribution outputs are employed to learn features explicitly internalizing label interdependencies using an RNN (*i.e., interdependency representation learning*). The latter foresees both, the consideration of interdependencies among the sequence of individual class probability values (CLIM) as well as of interdependencies among the individual task-wise class label probability distributions (TLIM).

Looking at computational costs, building on label probability sequences keeps interdependency modeling sparse, and training time consumption remains favorable compared to STL. At the same time, it facilitates the proposed MTL framework genericity for the expansion toward additional classification tasks.

The experimental evaluations showed that the presented MTL models consistently outperform their STL and hard parameter sharing MTL counterparts. Thereby, they allow for obtaining accuracy gains across all individual tasks. Comparing *multitask stacking* and label *interdependency representation learning* the results indicate, that the latter allows for an enhanced capturing of cross-task label interdependencies. Resulting accuracy measures further revealed that the two label interdependency modeling strategies *multitask stacking* and *interdependency representation learning* allow for synergies. Hence, for both frontend CNN architectures, *DenseNet121* and *EfficientNetV2-B3* a combination of both attained the highest accuracy estimates. EfficientNetV2-B3 CLIM-MS_{BL} with $\Delta_m(mean)$ values of +4.80% OA and +6.73% κ resulted in the highest generalization capability estimates overall and allowed for the joint predictions of the five tasks (*i.e., height*, *LLRS material*, *SBST*, *roof shape,* and *block position* classification) with mean cross-task accuracy values of 88.03% OA and 83.96% κ.

In parallel, it is confirmed that training time consumption compared to STL can be substantially reduced. Hence, given our addressed application and data – multicriteria building characterization with street-level imagery – the proposed MTL architectural framework allows to exploit the full potential of MTL, *i.e.*, improved generalization capabilities coupled with reduced training time consumption, for a considerable number of classification tasks.

To provide a comprehensive picture on the vulnerability of buildings exposed to multiple natural hazards using street-level imagery in a generic and efficient fashion, we suggest to explore the potential of the proposed MTL framework to consider further target variables in the future.

Acknowledgments

This study has been conducted as part of the projects RIESGOS (grant no. 03G0876A) and RIESGOS 2.0 (03G0905A), funded by the German Federal Ministry of Education and Research (BMBF).

References

Allen, L., Charleson, A. W., Brzev, S., Scawthorn, C., 2023. Glossary for GEM taxonomy, Ver. 4.0.0. URL: https://taxonomy.openquake.org/ (accessed on 27 July 2023).

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J., 2010. Google street view: capturing the world at street level. Computer 43, 32–38.

Aravena Pelizari, P., Spröhnle, K., Geiß, C., Schoepfer, E., Plank, S., Taubenböck, H., 2018. Multi-sensor feature fusion for very high spatial resolution built-up area extraction in temporary settlements. Remote Sens. Environ. 209, 793–807.

Aravena Pelizari, P., Geiß, C., Aguirre, P., Santa María, H., Taubenböck, H., 2021. Automated building characterization for seismic risk assessment using street-level imagery and deep learning. ISPRS J. Photogramm. Remote Sens. vol. 180, 370-386.

Biljecki, F., Ito, K., 2021. Street view imagery in urban analytics and GIS: A review. Landscape and Urban Planning 215, 104217.

Buda, et al., 2018. A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks 106, 249–259.

Caruana, R., 1997. Multitask Learning. Machine Learning, 28, 41–75.

Charte, F.et al., 2015. Addressing imbalance in multilabel classification: Measures and random resampling algorithms, Neurocomputing 163, 3-16.

Chen B., Feng Q., Niu B., Yan F., Gao B., Yang J., Gong J., Liu J., 2022. Multi-modal fusion of satellite and street-view images for urban village classification based on a dual-branch deep neural network. Int. J. Appl. Earth Obs. Geoinf. 109, 102794.

Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A., 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. International Conference on Machine Learning, 793–802.

Cheng, G., Xie, X., Han, J., Guo, L., Xia, G., 2020. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 3735–3756.

Cho, K, Van Merriënboer, B., Gülçehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proc. of the Conf. on Empirical Methods in Natural Language Processing, 1724–1734.

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv eprint arXiv:1412.3555.

Dai, J., He, K., Sun, J., 2016. Instance-aware semantic segmentation via multi-task network cascades. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3150–3158.

Esch, T., Heldens, W., Hirner, A., Keil, M., Marconcini, M., Roth, A., Zeidler, J., Dech, S., Strano, E., 2017. Breaking new ground in mapping human settlements from space–The Global Urban Footprint. ISPRS J. Photogramm. Remote Sens. 134, 30–42.

Esquivel-Salas, L.C., Schmidt-Díaz, V., Pittore, M., Hidalgo-Leiva, D., Haas, M., Moya-Fernández, A., 2022. Remote structural characterization of thousands of buildings from San Jose, Costa Rica. Front. Built Environ. 8:947329.

Garreton, M., 2017. City profile: Actually existing neoliberalism in Greater Santiago. Cities 65 (2017), 32–50.

Geiß, C., Taubenböck, H., 2013. Remote sensing contributing to assess earthquake risk: from a literature review towards a roadmap. Nat. Hazards 68 (1), 7–48.

Geiß, C., Aravena Pelizari, P., Marconcini, M., Sengara, W., Edwards, M., Lakes, T., Taubenböck, H., 2015. Estimation of seismic building structural types using multisensor remote sensing and machine learning techniques. ISPRS J. Photogramm. Remote Sens. 104, 175–188.

Geiß, C., Thoma, M., Pittore M., Wieland, M., Dech, S., and Taubenböck, H., 2017a. Multitask Active Learning for Characterization of Built Environments with Multisensor Earth Observation Data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10(12), 5583–5597.

Geiß, C., Aravena Pelizari, P., Schrade, H., Brenning, A., Taubenböck, H., 2017b. On the effect of spatially non-disjoint training and test samples on estimated model generalization capabilities in supervised classification with spatial features. IEEE Geosci. Remote Sens. Lett. 14 (11), 2008–2012.

Geiß, C., Aravena Pelizari, P., Blickensdörfer, L., Taubenböck, H., 2019. Virtual Support Vector Machines with self-learning strategy for classification of multispectral remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 151, 42–58.

Geiß, C., Zhu, Y., Qiu, C., Mou, L., Zhu, X. X., and Taubenböck, H., 2022a. Deep Relearning in the Geospatial Domain for Semantic Remote Sensing Image Segmentation, IEEE Geosci. Remote Sens. Lett., 19, 1–5, 8002705.

Geiß, C., Brzoska, E., Aravena Pelizari, P., Lautenbach, S., Taubenböck, H., 2022b. Multi-target Regressor Chains with Repetitive Permutation Scheme for Characterization of Built Environments with Remote Sensing. Int. J. Appl. Earth Obs. Geoinf. 106, 102657.

Geiß, C., Priesmeier, P., Aravena Pelizari, P. et al., 2022c. Benefits of global earth observation missions for disaggregation of exposure data and earthquake loss modeling: evidence from Santiago de Chile. Nat. Hazards.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9, 249-256.

Godbole, S., Sarawagi, S., 2004. Discriminative methods for multi-labeled classification. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 22–30.

Gomez Zapata, J. C., Brinckmann, N., Harig, S., Zafrir, R., Pittore, M., Cotton, F., and Babeyko, A., 2021. Variable-resolution building exposure modelling for earthquake and tsunami scenario-based risk assessment: an application case in Lima, Peru, Nat. Hazards Earth Syst. Sci., 21, 3599–3628.

Gómez Zapata, J. C., Pittore, M., Brinckmann, N., Lizarazo-Marriaga, J., Medina, S., Tarque, N., and Cotton, F., 2023. Scenario-based multi-risk assessment from existing single-hazard vulnerability models. An application to consecutive earthquakes and tsunamis in Lima, Peru, Nat. Hazards Earth Syst. Sci., 23, 2203–2228.

Gonzalez, D., Rueda-Plata, D., Acevedo, A.B., Duque, J.C., Ramos-Pollán, R., Betancourt, A., García, S., 2020. Automatic detection of building typology using deep learning methods on street level images. Build. Environ. 177, 106805.

Gülçehre, C., Bengio, Y., 2016. Knowledge Matters: Importance of Prior Information for Optimization. Journal of Machine Learning Research 17, 1-32.

Hang, R., Liu, Q., Hong, D., Ghamisi, P, 2019. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification, IEEE Trans. Geosci. Remote Sens. 57 (8), 5384-5394.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification, 2015 IEEE International Conference on Computer Vision (ICCV), 1026-1034.

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

Hoffmann, E. J., Abdulahhad, K., Zhu, X. X., 2023. Using Social Media Images for Building Function Classification. Cities 133, 104107.

Hua, Y., Mou, L., Zhu, X. X., 2019. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification, ISPRS J. Photogram. Remote Sens., vol. 149, 188–199.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., 2017. Densely Connected Convolutional Networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2261-2269.

Ibrahim, M. R., Haworth, J., Cheng, T., 2020. Understanding cities with machine eyes: A review of deep computer vision in urban analytics. Cities, 96, 102481.

INE, 2018. Base Cártografica Censal. Alcances y consideraciones para el usuario. Departamento de Demografía y Censos, Instituto Nacional de Estadísticas, Chile.

Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures. Proc. 32nd Int. Conf. Mach. Learn. (ICML), 2342–2350.

Kang, J., Körner, M., Wang, Y., Taubenböck, H., Zhu, X. X., 2018. Building instance classification using street view images. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 44–59.

Kendall, A., Gal, Y., Cipolla, R., 2018. Multi-task learning using uncertainty to weight losses for scene geometry and semantics. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7482-7491.Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., 2015. Deeply-Supervised nets. Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, 562–570.

Liu, S., Johns, E., Davison, A.J., 2019. End-to-end multi-task learning with attention. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1871-1880.

Liu, S., Shi, Q., 2020. Multitask Deep Learning with Spectral Knowledge for Hyperspectral Image Classification. IEEE Geosci. Remote Sens. Lett. 17 (12), 2110–2114.

Long, M., Cao, Z., Wang, J., Yu, P. S., 2017. Learning multiple tasks with multilinear relationship networks. Advances in Neural Information Processing Systems 30 (NIPS 2017), 1593–1602.

Misra, I, Shrivastava, A, Gupta, A, Hebert, M., 2016. Cross-stitch networks for multi-task learning. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3994-4003.

Mou, L., Ghamisi, P., Zhu, X. X., 2017. Deep recurrent neural networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. 55 (7), 3639–3655.

Pan, S.J., Yang, Q., 2010. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Pittore, M., Wieland, M., Fleming, K., 2017. Perspectives on global dynamic exposure modelling for geo-risk assessment. Nat. Hazards 86 (Suppl 1), 7–30.

Read, J., Pfahringer, B., Holmes, G., Frank, E., 2011. Classifier chains for multi-label classification. Mach. Learn. 85 (3), 333–359.

Ruder, S., 2017. An overview of multi-task learning in deep neural networks. arXiv eprint arXiv:1706.05098.

Rueda-Plata, D., González, D., Acevedo, A.B., Duque, J.C., Ramos-Pollán, R., 2021. Use of deep learning models in street-level images to classify one-story unreinforced masonry buildings based on roof diaphragms. Build. Environ. 189, 107517.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, Li, 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115, 211–252.

Santa María, H., Hube, M. A., Rivera, F., Yepes-Estrada, C., Valcárcel, J. A., 2017. Development of national and local exposure models of residential structures in Chile. Nat. Hazards 86, 55–79

Schuster, M., Paliwal, K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45, 2673–2681.

Silva, V., Brzev, S., Scawthorn, C. et al., 2022. A Building Classification System for Multi-hazard Risk Assessment. Int. J. Disaster Risk Sci. 13, 161–177.

Spyromitros-Xioufis, E., Groves W., Tsoumakas G., Vlahavas I.., 2012. Multi-label classification methods for multi-target regression. arXiv eprint arXiv:1211.6581v1.

Sun, M., Zhang, F., Duarte, F., Ratti, C. 2022. Understanding architecture age and style through deep learning, Cities, 128, 103787.

Tan, M., Le, Q. V., 2021. EfficientNetV2: Smaller Models and Faster Training. Proceedings of the 38th International Conference on Machine Learning (ICML 2021), 139, 10096–10106.

Taubenböck, H., Münich, C., Zschau, J., Roth, A., Dech, S., Mehl, H., 2009. Assessing building vulnerability using synergistically remote sensing and civil engineering. In: Krek, Rumor, Zlatanova & Fendel (eds). Urban and Regional Data Management, Taylor & Francis Group, London, ISBN 978-0-415-55642-2, 287-300.

Ting, K. M., Witten, I.H., 1999. Issues in Stacked Generalization. Journal of Articial Intelligence Research 10, 271-289.

Vandenhende, S., Georgoulis, S., Van Gool, L., 2020. MTI-Net: Multi-scale task interaction networks for multi-task learning. European Conference on Computer Vision, 527–543.

Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L., 2022. Multi-Task Learning for Dense Prediction Tasks: A Survey, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (7), 3614-3633.

Wieland, M., Pittore, M., Parolai, S., Zschau, J., Moldobekov, B., Begaliev, U., 2012. Estimating building inventory for rapid seismic vulnerability assessment: Towards an integrated approach based on multisource imaging. Soil. Dyn. Earthq. Eng. 36, 70–83.

Wolpert, D. H., 1992. Stacked generalization, Neural Netw. 5 (2), 241-259.

Xu, D., Ouyang, W., Wang, X., Sebe, N., 2018. PAD-net: Multi-tasks guided prediction-anddistillation network for simultaneous depth estimation and scene parsing. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 675–684.

Yang, Y., Hospedales, T., 2017. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. Proc. Int. Conf. Learn. Representations 2017.

Yu, Q., Wang, C., McKenna, F. et al., 2020. Rapid visual screening of soft-story buildings from street view images using deep learning classification. Earthq. Eng. Eng. Vib. 19, 827–838.

Zhang, Y., Yang, Q., 2022. A Survey on Multi-Task Learning, IEEE Trans. Knowl. Data Eng. 34 (12), 5586-5609.

Zhang, F., Wu, L., Zhu, D., Liu, Y. 2019a. Social sensing from street-level imagery: A case study in learning spatio-temporal urban mobility patterns. ISPRS J. Photogramm. Remote Sens. 153, 48–58.

Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., Yang, J., 2019b. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4106-4115.

Zheng, X., Gong, T., Li, X., Lu, X., 2022. Generalized scene classification from small-scale datasets with multitask learning, IEEE Trans. Geosci. Remote Sens. 60, 1-11.

Zhu, Y., Geiß, C., So. E., Jin, Y., 2021. Multi-temporal Relearning with Convolutional LSTM Models for Land Use Classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 3251–3265.