Selecting Autoencoder Features for Layout Analysis of Historical Documents

Hao Wei
DIVA group
University of Fribourg
Bd. de Pérolles 90,
1700 Fribourg, Switzerland
hao.wei@unifr.ch

Mathias Seuret
DIVA group
University of Fribourg
Bd. de Pérolles 90,
1700 Fribourg, Switzerland
mathias.seuret@unifr.ch

Kai Chen
DIVA group
University of Fribourg
Bd. de Pérolles 90,
1700 Fribourg, Switzerland
kai.chen@unifr.ch

Andreas Fischer
DIVA group
University of Fribourg
and iCoSys Institute
University of Applied Sciences
and Arts Western Switzerland
andreas.fischer@unifr.ch

Marcus Liwicki
DIVA group
University of Fribourg
Bd. de Pérolles 90,
1700 Fribourg, Switzerland
marcus.eichenberger-liwicki@unifr.ch

Rolf Ingold
DIVA group
University of Fribourg
Bd. de Pérolles 90,
1700 Fribourg, Switzerland
rolf.ingold@unifr.ch

ABSTRACT

Automatic layout analysis of historical documents has to cope with a large number of different scripts, writing supports, and digitalization qualities. Under these conditions, the design of robust features for machine learning is a highly challenging task. We use convolutional autoencoders to learn features from the images. In order to increase the classification accuracy and to reduce the feature dimension, in this paper we propose a novel feature selection method. The method cascades adapted versions of two conventional methods. Compared to three conventional methods and our previous work, the proposed method achieves a higher classification accuracy in most cases, while maintaining low feature dimension. In addition, we find that a significant number of autoencoder features are redundant or irrelevant for the classification, and we give our explanations. To the best of our knowledge, this paper is one of the first investigations in the field of image processing on the detection of redundancy and irrelevance of autoencoder features using feature selection.

CCS Concepts

•Computing methodologies → Image segmentation; Feature selection; •Applied computing → Document analysis;

Keywords

Historical documents; layout analysis; autoencoders; selected features; accuracy; feature dimension

HIP '15, August 22 2015, Nancy, France

DOI: http://dx.doi.org/10.1145/2809544.2809548

1. INTRODUCTION

Historical documents are one of the most challenging types of documents for automatic document images analysis (DIA), because they encompass a large number of different writing supports, writing instruments, scripts, and digitalization qualities. Furthermore, old documents often suffer from degradation and may contain decorations that interfere with other document contents. In the pipeline of automatic DIA, layout analysis is an important prerequisite for further stages. It aims at splitting a document image into regions of interest.

Considering the large variety of historical document images, machine learning is a promising approach to tackle the problem of layout analysis. Following recent advances in machine learning, a promising strategy is to use training data not only for training classifiers based on hand-crafted features but also for learning the features automatically, for example using convolutional autoencoders (CAE) [11]. In [2], a layout analysis system is introduced that follows this approach and relies on CAE features to classify each pixel into four basic layout classes, namely periphery, background, text block, and decoration.

CAE features aim to find a low-dimensional representation of the document images, from which the original image can be reconstructed with a high quality. That is, the dimensionality of the features is optimized with respect to the task of reconstruction. However, it remains unclear whether this dimensionality is optimal for the subsequent classification task.

In this paper, we propose a novel feature selection method called ASFS-AGA working on CAE features, to increase the classification accuracy, reduce the feature dimension, and investigate redundancy and irrelevance of CAE features. The proposed ASFS-AGA is based on our previous work [18]. The method is a cascade of adapted versions of two conventional methods: sequential forward selection (SFS) [9] and genetic algorithm (GA) [20]. ASFS extends SFS to incorporate more predictive features, and AGA takes into account both accuracy and number of features to refine the features

obtained by ASFS. We evaluate ASFS-AGA on UCI Machine Learning Repository [12] and on IAM-HistDB [5, 6] datasets for layout analysis of historical documents. The experiments demonstrate that ASFS-AGA is competitive with respect to both of the accuracy and the number of selected features.

We point out that a significant number of autoencoder features are redundant or irrelevant when they are used for the classification, in the context of our layout analysis task. Although there exist works about feature selection of handcrafted features, e.g., Gabor features selection [15, 17], and local binary patterns selection [7, 21], there is hardly any investigation about feature selection of autoencoders features. To the best of our knowledge, this paper provides one of the first investigations in the field of image processing on the detection of redundancy and irrelevance of autoencoder features using feature selection.

2. RELATED WORKS

Feature selection methods use a search technique to propose feature subsets, and use an evaluation measure to evaluate the feature subsets [9]. The search techniques include sequential forward selection (SFS), sequential backward selection (SBS), genetic algorithm (GA), etc. SFS adds more and more good features starting with an empty feature subset. SBS removes more and more bad features starting with the complete feature set. GA is based on natural selection and uses a fitness function to decide the better feature subsets during the evolvement. According to the evluation measure, feature selection methods are divided into three categories: filters, wrappers, and embedded methods. Filters evaluate a feature subset by calculating the relationship between features and labels, e.g., mutual information and correlation. Wrappers evaluate a feature subset by computing the cross-validation accuracy of the features. Compared to filters, wrappers usually achieve higher accuracy, but are more time-consuming. Embedded methods embed feature selection in the process of machine learning. For more details about feature selection methods, please refer to [9].

There exist some variations of SFS. Linear Forward Selection [8] limits the number of features to be considered at each step of SFS, in order to speed up the selection. In our previous work, we proposed in [18] a feature selection method call ASFS-GA[1] for the layout analysis of historical document. ASFS-GA is a search technique and was implemented as a wrapper in [18]. ASFS-GA adapts SFS in order to include more predictive features, then GA is used to refine the feature subset. The method achieves competitive performance compared to several conventional methods. ASFS-GA was implemented as a filter in [19]. Experiments in [18] and [19] demonstrated that some features, e.g., local binary patterns, are significantly more predictive than other features for the layout analysis of historical documents.

There exist some variations of GA as well. Tan et al. [16] proposed a GA with a fitness function taking into account both the accuracy and the number of selected features. Their experiments demonstrate that the proposed method is robust and effective to find feature subsets with higher accuracy and/or less features compared to other methods. Some other GAs for feature selection with similar fitness functions were proposed in [14] for image annotation, and in [1] for target detection in images.

Feature learning by autoencoders has increasingly been used in the recent years. However features learned by autoencoders may be somewhat redundant or irrelevant for the classification. To solve the problem, Zhou et al. [23] proposed an incremental feature learning algorithm. The algorithm merges similar features which are considered redundant to produce more compact feature representations. Chen et al. [4] used autoencoders to learn interest point descriptors (features). They used mean pooling to remove redundant components of a descriptor.

Since feature selection is a process to remove redundant and irrelevant features in nature, using feature selection on autoencoder features is straightforward and may be promising. However, there are very few publications working on it. Zhao et al. [22] used autoencoders to map high-dimensional sparse features to low-dimensional features, followed by feature selection. The selected features are used for statistical machine translation. But the authors neither gave details of their feature selection method, nor further investigated the selected features. In the general field of image processing and also document image analysis, feature selection of autoencoder features is still missing.

3. SYSTEM OVERVIEW

On historical document images, we perform feature extraction and feature selection, train classifiers, and finally classify pixels on unseen images into *periphery*, *background*, *text block*, and *decoration*.

3.1 Historical document images

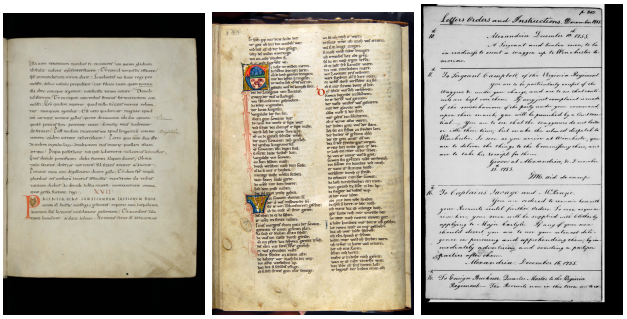
The images we are working on are from IAM-HistDB (IAM Historical Document Database)[2]. IAM-HistDB contains three handwritten historical datasets: Saint Gall dataset, Parzival dataset, and Washington dataset [5, 6]. The images on the three datasets are of different nature. The first two datasets consist of images of medieval manuscripts written with ink on parchment and the images are in color, while Parzival dataset suffers from many degradations. Washington dataset consists of images of manuscript written with ink on paper and the images are in gray levels. Since the images are of high resolution, we resize them to smaller sizes to reduce computational cost. Three examples of resized images from the datasets are given in Fig. 1.

3.2 Feature extraction and selection

We use the feature learning method presented in [2] to extract features. The method uses a convolutional autoencoder (CAE) which is based on the autoencoder architecture [11]. An autoencoder (AE) is an artificial neural network where the input and the desired output are the same. It uses backpropagation algorithm to train the network until the error between the actual and desired output is less than a threshold. After the training, the activations of the hidden layer form the learned features. In [2], the authors use a CAE comprising three-level AEs to learn features. The CAE learn features of a pixel from details to big shapes level by level. The first-level AE learns from a small patch (5×5) around the pixel. The learned features reflect small

[1]Note that in [18] we didn't call the method ASFS-GA, but in this paper we name it for the simplicity.

[2]http://www.iam.unibe.ch/fki/databases/iam-historical-document-database



(a) (b) (c)

Figure 1: Page examples from the three datasets. They are from Saint Gall dataset (Cod. Sang. 562, page 62, Abbey Library of St. Gall, (SG30)), Parzival dataset (Cod. 857, page 144, Abbey Library of St. Gall, (PAR23)), and Washington dataset (G. W. Papers, page307, Library of Congress, (GW10)) respectively.

details around the pixel. The second-level AE learns from a medium patch (15×15) around the pixel. Finally the third-level AE learns from a big patch (35×35) around the pixel. The features learned on the third level represent the contour of the shape in the big patch. The dimensions of learned features from each level are 80, 60, and 40 respectively. These features are concatenated together to form the final feature vector (180d). Please refer to [2] for details of the feature learning strategy.

Since the third-level AE learns from 35×35 patches, pixels on the border of the images are not taken into account for the classification. Thus the training and testing pixels are from the resized images which are further trimmed by 17 pixels on the border. In addition, since borders are trimmed, areas of periphery on images from Parzival and Washington datasets are entirely removed. Furthermore, since Washington dataset has no decoration class, it has only two classes left. Details of the data for the classification is shown in Table 1.

Table 1: Data for the classification

	# training pixels	# testing pixels	# features	# classes
Saint Gall	175,239	250,470	180	4
Parzival	330,480	179,010	180	3
Washington	183,340	91,670	180	2

The data detailed in Table 1 is then used for feature selection. Our proposed feature selection method is elaborated in the next section.

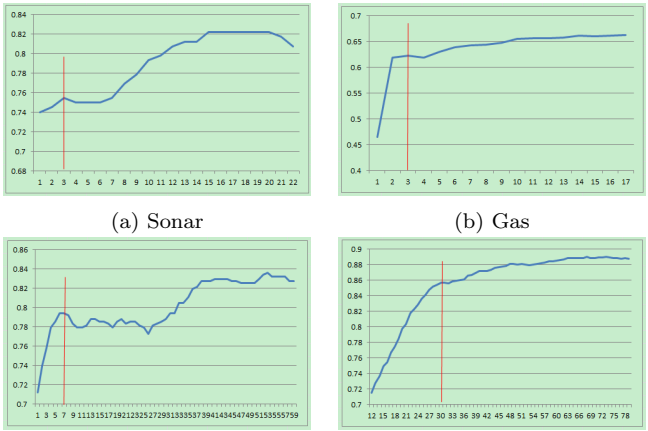
Finally we use the trained classifier to classify pixels on unseen pages into four classes for Saint Gall datasets, three classes for Parzival dataset, and two classes for Washington dataset. The ground truth of the page examples in Figure 1 is shown later in Figure 4a, Figure 4d, and Figure 4g respectively. Details of the ground truth are explained in [3]. Note that the ground truth in Figure 4a, Figure 4d, and Figure 4g represents the resized images which are further trimmed. Thus borders of images are not visible. In Figure 4a, the area of periphery at the bottom is tiny because of the trimmed image.

4. PROPOSED FEATURE SELECTION

We propose a novel feature selection method ASFS-AGA. The method is a cascade of ASFS (adapted sequential forward selection) and AGA (adapted genetic algorithm). We implement the method as a wrapper, and use the Naive Bayes classifier to evaluate the cross-validation accuracy of feature subsets.

4.1 Adapted sequential forward selection

Sequential Forward Selection (SFS) iteratively adds the best feature from the remaining unselected features to the feature subset, until the cross-validation accuracy of the new feature subset becomes worse than that of the last subset. However, SFS is deficient. We demonstrate our argument in Fig. 2 on four datasets from the UCI repository [12], a commonly used repository for the machine learning community.



(a) Sonar (b) Gas

(c) Musk1 (d) Semeion

Figure 2: Sequential forward selection on four UCI datasets (Sonar, Gas, Musk1, and Semeion). X axis represents the number of selected features. Y axis represents the cross-validation accuracy of the dataset represented by selected features. The red vertical line indicates where SFS stops. The blue curve indicates how the cross-validation accuracy changes if we force SFS to keep adding new best features even after SFS is supposed to stop.

Fig. 2 shows that if we force SFS to keep adding new features regardless of how the accuracy changes, the accuracy reaches a new higher level at some point. This discovery gives us an inspiration to achieve higher accuracy. The change of the accuracy can be summarized to Fig. 3.

We thus propose a method called Adapted Sequential Forward Selection (ASFS), based on our previous work ASFS-GA [18]. It forces SFS to continue adding new best features until the number of selected features reaches a previously fixed number q. We don't give fixed rule about how to choose q, since we don't know when the accuracy reaches the highest level, unless ASFS runs to the last iteration (i.e., ASFS includes all of the features). In our experiments (Section 5), normally q is greater than p which is the number of features selected by SFS, and less than two-thirds of the total number of features. We use $subset_{ASFS}$ to denote the feature subset by ASFS.

During the execution of ASFS and also SFS, if a feature is selected, it will never be removed. However, a drawback is that this feature may be redundant or less predictive if it

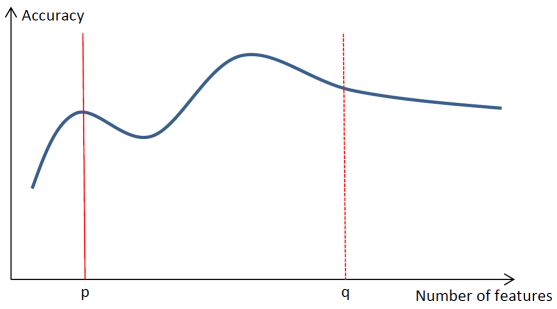


Figure 3: Adapted sequential forward selection (ASFS). p is where SFS stops. q is where ASFS stops.

is combined with some features selected later. This drawback has been discovered by our previous work [18, 19]. We thus use another feature selection method below to refine $subset_{ASFS}$.

4.2 Adapted genetic algorithm

We propose another method called Adapted Genetic Algorithm (AGA) to refine $subset_{ASFS}$. Before elaborating AGA, we give a short description about the genetic algorithm (GA) below.

Genetic algorithm is a technology based on the process of natural selection. Initially, a generation contains some individuals which are randomly initialized. In the context of feature selection, an individual of each generation is represented by a string consisting of 1s and 0s, where the 1 at position i means that the feature i is selected, and the 0 means the feature is not selected. The individuals use operations including crossover, mutation, and elitist selection to generate the next generation. For the wrapper category, the fitness of each individual of GA is the cross-validation accuracy of the dataset represented by the selected features. GA iterates until it reaches a fixed number of generations, or the best individual doesn't change any more in the successive generations. For detailed description, please refer to [20].

AGA changes the fitness of GA to a combination of the accuracy and the number of selected features. The fitness function we use is the same with that proposed in [16]. The fitness function is formulated below:

$$fitness = w * c(subset) + (1 - w) * (1/s(subset))$$

where $subset$ is the feature subset represented by an individual within a generation, $c(subset)$ is the cross-validation accuracy of the dataset represented by $subset$, $s(subset)$ is the size of $subset$, and w is the weight between 0 and 1. The $fitness$ is proportional to the accuracy and inversely proportional to the number of selected features. By adjusting w, we could control the priority between the accuracy and the number of selected features. If w is high, we prefer the feature subset with higher accuracy. Otherwise, we prefer the feature subset with less features. When $w = 1$, AGA becomes the standard GA.

$subset_{ASFS}$ is refined by AGA. The resultant feature subset is then used for training classifiers and testing.

5. EXPERIMENTS

We apply ASFS-AGA on UCI datasets and the layout analysis datasets in Table 1. For simplicity, in the following we call the datasets in Table 1 HisDoc datasets[3]. We also compare our method with our previous work ASFS-GA [18] and three conventional methods, i.e., sequential forward selection (SFS), sequential backward selection (SBS), and genetic algorithm (GA). The reason why we don't compare ASFS-AGA with the method proposed in [19] is that the method in [19] is a filter, which is of different category of feature selection methods from ASFS-AGA (wrapper).

As elaborated in Section 4, we need to set the parameter q to control where ASFS stops, and the weight w of AGA. To make a fair evaluation of ASFS-AGA, we test 3 random values for q according to the way that we explained in the Section 4.1 to choose q, and 10 values for w from 0.90 to 0.99 with increment of 0.01. Then we obtain the mean and the best accuracy of the 30 (3×30) combinations. Since our method is based on our previous method ASFS-GA, we also use the same values q to evaluate ASFS-GA, and obtain the mean and the best accuracy of the 3 tests. In this way, we guarantee that the comparison between ASFS-AGA and ASFS-GA is fair. Our implementation is based on the source code of WEKA Data Mining Software [10]. For SFS, SBS, and GA, all parameters are set by default by WEKA, except that the number of generations is set to 2000 for GA. For ASFS-GA and ASFS-AGA, the number of generations is also set to 2000 for the GA component, and other parameters except q and w are set by default by WEKA.

In the experiments below, the cross-validation accuracy indicates the 10-fold cross-validation accuracy using the corresponding features, and testing accuracy indicates the accuracy on the testing datasets using the corresponding features. We use the Naive Bayes classifiers for all experiments.

5.1 Results on UCI datasets

We first evaluate the feature selection methods on the UCI datasets [12]. In order to test the robustness of ASFS-AGA, 10 datasets (Sonar, Libras, Gas, Musk1, Musk2, Semeion, LSVT, Madelon, Isolet, and Multiple Features) are picked up from different domains. The 10 datasets are miscellaneous. They range in number of features from 60 to 649, in number of classes from 2 to 26, and in number of instances from 126 to 13,910. The statistics of the 10 datasets are shown in Table 2.

Note that we directly use the UCI datasets for feature selection, and no autoencoder is applied on the datasets. Since for most of the datasets, the testing dataset is not available, we use cross-validation accuracy and the dimension of selected features to evaluate feature selection methods. The result is shown in Table 3. In the Table 3, ACC stands for the cross-validation accuracy, and DIM stands for the dimension. We give both of the mean and the best accuracy for ASFS-GA and ASFS-AGA. ASFS-AGA $\star$ means the dimension of the feature subset with the best accuracy by ASFS-AGA. The same applies to ASFS-GA $\star$.

Table 3 shows that ASFS-AGA is always capable of achieving the best accuracy, given proper parameters shown in Table 4. The mean accuracy of ASFS-AGA is comparable with that of ASFS-GA, and is better than SFS, SBS, and GA in general. The dimension of the feature subset by ASFS-AGA in general is lower than those by SBS and GA, comparable with that by ASFS-GA, and higher than that by SFS.

[3]We name the datasets after our HisDoc project, http://diuf.unifr.ch/main/hisdoc/.

Table 2: Number of features, classes, and instances of the 10 UCI datasets

	Sonar	Libras	Gas	Musk1	Musk2	Semeion	LSVT	Madelon	Isolet	Multi-Features
# Features	60	90	128	166	166	256	309	500	617	649
# Classes	2	15	6	2	2	10	2	2	26	10
# Instances	208	360	13910	476	6598	1593	126	2000	1559	2000

Table 3: Accuracy and dimension of feature sets on the 10 UCI datasets

		Sonar	Libras	Gas	Musk1	Musk2	Semeion	LSVT	Madelon	Isolet	Multi-Features
ACC (%)	Full features	67.79	62.78	57	75.21	83.86	85.94	56.35	58.53	83.77	95.35
	SFS	75.48	68.06	62.25	79.41	91.48	85.69	94.44	63.48	90.06	98.15
	SBS	75.48	65.56	65.16	82.56	87.44	88.26	80.95	61.28	85.82	97.35
	GA	84.62	69.44	66.43	84.87	91.35	89.2	67.46	64.33	88.77	97.25
	ASFS-GA mean	84.61	69.54	66.9	85.85	92.97	88.93	95.50	63.78	92.75	99.13
	ASFS-GA best	**85.58**	69.72	67.10	86.13	93.73	89.39	96.83	64.08	93.01	99.2
	ASFS-AGA mean	84.6	69.66	67.03	85.82	92.93	89.12	94.9	64.1	92.91	99.08
	ASFS-AGA best	**85.58**	**70.28**	**67.22**	**86.97**	**93.92**	**90.27**	**97.62**	**64.68**	**93.52**	**99.3**
DIM	Full features	60	90	128	166	166	256	309	500	617	649
	SFS	**3**	**11**	**3**	**7**	**7**	**30**	6	**6**	**28**	**13**
	SBS	44	81	52	81	134	199	**3**	483	550	187
	GA	17	38	25	63	67	134	64	258	276	238
	ASFS-GA mean	15	22	20	44	37	62	20	78	108	76
	ASFS-GA ⋆	18	20	13	33	35	73	15	42	65	72
	ASFS-AGA mean	14	26	15	32	35	61	20	80	102	85
	ASFS-AGA ⋆	18	19	13	34	36	73	14	39	96	55

Table 4: Parameters resulting in the best accuracy of ASFS-AGA on the 10 UCI datasets

	Sonar	Libras	Gas	Musk1	Musk2	Semeion	LSVT	Madelon	Isolet	Multi-Features
q of ASFS	40	35	60	100	80	120	50	100	200	200
w of AGA	0.96	0.99	0.95	0.97	0.97	0.91	0.95	0.98	0.95	0.98

5.2 Results on HisDoc datasets

In this section, we give classification result, segmentation result, and analysis of selected features.

5.2.1 *Classification result*

As we mentioned in Section 2, wrapper is a time-consuming category of feature selection, especially for HisDoc datasets consisting of large numbers of pixels (instances). If we use all instances for feature selection, it will cost 1-2 weeks. Thus we randomly chose 10%, 5%, and 10% of pixels from the three datasets respectively to perform feature selection. After feature selection, these same subsets of instances are used to train classifiers.

The result of feature selection is shown in Table 5. Similar with Table 3, the mean and the best cross-validation and testing accuracies of ASFS-GA and ASFS-AGA are given. Regarding testing accuracy, ASFS-AGA ⋆ means testing accuracy using features achieving the best cross-validation accuracy. Regarding dimension, ASFS-AGA ⋆ means number of features achieving the best cross-validation accuracy. The same applies to ASFS-GA ⋆.

Table 5 shows ASFS-AGA is always capable of achieving the best cross-validation accuracy, given proper parameters shown in Table 6. With the features achieving the best cross-validation accuracy, ASFS-AGA achieves the second best testing accuracy. The reason why the best cross-validation accuracy doesn't lead to the best testing accuracy could be overfitting. Because we use only a small subset of instances (pixels) to select features and train classifiers, this may account for overfitting. With respect to selected features, the selected features by ASFS-AGA are fewer than those by SBS, GA, and ASFS-GA in general, and more than those by SFS.

Table 7 shows the result of a t-test comparing the mean performances of ASFS-AGA and ASFS-GA, with respect to cross-validation (CV) accuracy, testing accuracy, and dimension of selected features. The mean performances of ASFS-AGA and ASFS-GA refer to those shown in Table 5. The t-test shows that ASFS-AGA is in general better than ASFS-GA with high confidence. The low confidence of testing accuracy on Parzival dataset may be due to the fact that there are only three evaluations of ASFS-GA.

No matter which feature selection method we use, the cross-validation and testing accuracies are increased, and feature dimensions are decreased. Regarding the best cross-validation accuracy or the best testing accuracy, the corresponding features are always fewer than 20% of full features. This demonstrates that a significant number of autoencoder features are redundant or irrelevant for the classification. This finding may remind researchers to generate more compact features when they are using autoencoders for feature learning.

Table 5: Accuracy and dimension of feature sets on HisDoc datasets

		Saint Gall	Parzival	Washington
Cross Validation Accuracy (%)	Full features	78.34	87.44	91.09
	SFS	96.11	93.25	92.32
	SBS	94.15	88.82	91.81
	GA	93.09	93.08	95.15
	ASFS-GA mean	96.13	93.66	95.47
	ASFS-GA best	96.27	93.80	95.53
	ASFS-AGA mean	96.21	93.78	95.49
	ASFS-AGA best	**96.69**	**93.97**	**95.63**
Testing Accuracy (%)	Full features	78.14	92.53	83.48
	SFS	**94.67**	94.68	**93.94**
	SBS	93.41	92.97	84.46
	GA	91.93	95.00	89.81
	ASFS-GA mean	94.23	95.27	90.53
	ASFS-GA ⋆	94.34	**95.33**	90.38
	ASFS-AGA mean	94.49	95.26	91.12
	ASFS-AGA ⋆	94.43	95.27	91.50
Dimension	Full features	180	180	180
	SFS	**12**	**18**	**5**
	SBS	22	166	163
	GA	25	46	64
	ASFS-GA mean	19	35	50
	ASFS-GA ⋆	18	31	53
	ASFS-AGA mean	14	31	40
	ASFS-AGA ⋆	**12**	32	33

Table 6: Parameters resulting in the best cross-validation accuracy of ASFS-AGA on HisDoc datasets

	Saint Gall	Parzival	Washington
q of ASFS	60	60	60
w of AGA	0.93	0.98	0.97

Table 7: Confidence (%) of assertion that ASFS-AGA is better than ASFS-GA on HisDoc dataset

	Saint Gall	Parzival	Washington
CV accuracy	69.33	89.75	67.80
Testing accuracy	93.19	34.01	96.38
Dimension	99.99	82.77	95.41

5.2.2 Segmentation result

The segmentation results of the page examples in Figure 1 are shown in Figure 4.

We observe that misclassified pixels can be generally divided into two categories. The first category of misclassified pixels are located on the boundary between each areas. On one hand, it is up to experts' personal opinions to define the class to which a pixel around the boundary belong, when they make the ground truth. Thus there is no strict definition about the ground truth of pixels around the boundary. On the other hand, since the features of pixels on the two sides of the boundary are quite similar, it is not surprising that some pixels are misclassified.

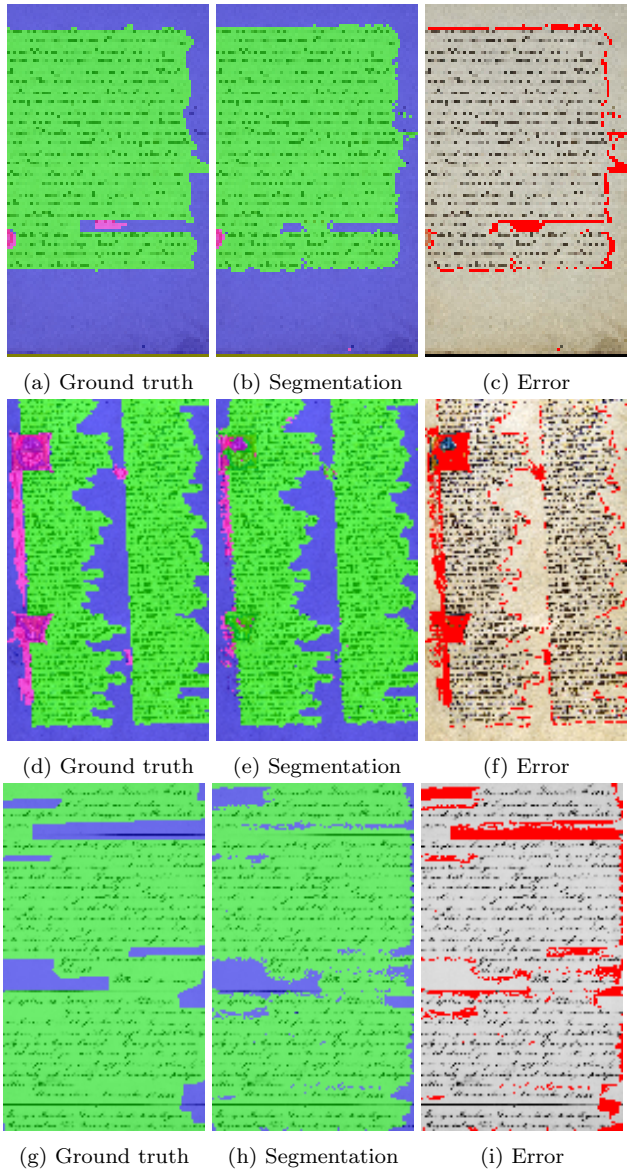
The second category of misclassified pixels are pixels of decoration which are misclassified as text block or background. For example, in Figure 4b a decoration area is completely misclassified as text block. In Figure 4e, some pixels of decoration are misclassified as either text block or background. In other word, the class of decoration is somewhat overwhelmed by the classes of text block and background on the segmentation images. This is the class imbalance problem, where some classes (major classes) have significantly more instances than other classes (minor classes) [13]. In this problem, the classifiers have a bias towards the major classes, and thus the instances belonging to the minor classes have higher probability to be misclassified than those belonging to the major classes [13]. The class of decoration is a minor class in Parzival and Saint Gall datasets. This is why it is not fully detected.

5.2.3 Analysis of selected features

In Figure 5, each cell of each image is a reconstructed patch by a single hidden unit (feature) of autoencoder. The selected features are highlighted with red rectangles. As we explained in Section 3.2, Level 1 encodes details of small patches, Level 2 encodes medium patches, and Level 3 captures the shape of big elements spanning over multiple text lines.

In order to differentiate between areas of periphery, text block, background, and decoration, features capturing big area around the pixel are important. Compared to Saint Gall images, Parzival and Washington images have higher resolution in height and width. In addition, the sizes of patches for feature learning on each level are fixed (5×5 for Level 1, 15×15 for Level 2, and 35×35 for Level 3).[4] This is why more features on Level 2 and Level 3 are selected on Parzival and Washington images, compared to Saint Gall images. In contrast, Saint Gall images have a lower resolution, and the layout is relatively simple (single text block, less and smaller decorations), thus features from Level 1 are almost capable of the classification.

[4]In this paper we use fixed sizes of patches. We could adjust the sizes of patches according to the resolution of the images, but will do it in future.



(a) Ground truth (b) Segmentation (c) Error

(d) Ground truth (e) Segmentation (f) Error

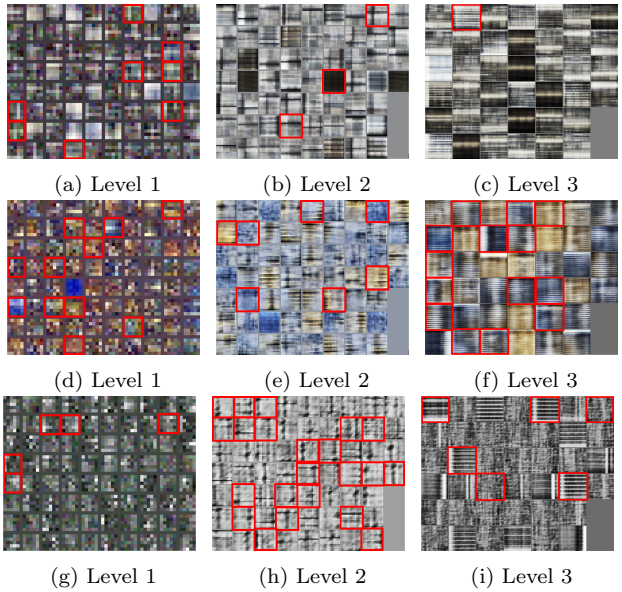
(g) Ground truth (h) Segmentation (i) Error

Figure 4: Ground truth, segmentation result, and segmentation error of the page examples from Figure 1. Borders of images are not visible due to the trimming. The three rows correspond to Figure 1a, Figure 1b, and Figure 1c respectively. The parts of text block, background, decoration, and periphery are painted with green, blue, magenta, and gray respectively on the images of ground truth and segmentation. On the images of error, the red indicates the pixels misclassified.

Parzival images have more decorations than images from the other two datasets. In order to detect the decorations, we think features capturing the decorations should be used for the classification. This is why many features capturing the decorations on Level 3 on Parzival dataset are selected. For example, the two red rectangles at the bottom of Figure 5f shows the rough shape of some decorations.

Compared to Parzival and Saint Gall dataset, Washington dataset has no decorations. Thus Level 3 shows the rough shape of the text and background. For example, the second red rectangle on the first row of Figure 5i shows the shape of text and its margin in the left.

The last observation is that intuitively good features for the reconstruction are not necessarily good for the classification. For example, some unselected features in Figure 5f shows the shape of decorations, which means these features are good for the reconstruction of the image, but these features are not selected for the classification. The reasons are twofold. First, the images generated by some unselected features are similar with those generated by some selected features, which means these unselected features are redundant with respect to the selected features. Since one aim of feature selection is to remove redundant features, it is straightforward that these redundant features are not selected. Second, we speculate that there is no explicit relation between good features for the reconstruction and good features for the classification. To the best of our knowledge, this is a new speculation, and worth further research.



(a) Level 1 (b) Level 2 (c) Level 3

(d) Level 1 (e) Level 2 (f) Level 3

(g) Level 1 (h) Level 2 (i) Level 3

Figure 5: Selected features marked with red rectangles on Saint Gall, Parzival, and Washington datasets. Each cell in each images is the reconstructed image by a single hidden unit (feature) of autoencoder. The three rows of images correspond to Saint Gall, Parzival, and Washington datasets respectively.

6. CONCLUSIONS

In this paper, we propose a hybrid feature selection method which cascades adapted versions of two conventional methods. Compared to conventional methods and our previous work, the proposed method is more flexible and robust. The experiment on UCI datasets demonstrates that the proposed method is always capable of achieving the best cross-validation accuracy. On our layout analysis datasets, the proposed method is always capable of achieving the best cross-validation accuracy, and the second best testing accuracy, with relatively less features.

Redundancy and irrelevance are important factors for any kind of features to be used for the classification. However in the literature, there is still a lack of research investigating how redundant or irrelevant autoencoder features are. Using feature selection methods, we reveal that in our layout analysis task, more than 80% of autoencoder features are

redundant or irrelevant. Our work may remind researchers to design autoencoders (e.g., number of hidden units) more carefully and to even increase the accuracy by removing redundant and irrelevant features.

Our future work includes three aspects. First, given proper parameters, ASFS-AGA always achieves the best cross validation accuracy on different kinds of datasets. However, these parameters are found by greedy tests on all parameter combinations. Thus finding the proper parameters in a more intelligent way is our concern. Second, in our experiment, to save the computational time, we used small subsets of instances to perform feature selection. As a result, overfitting occurred, and we did not achieve the best testing accuracy in spite of the best cross-validation accuracy. Since our datasets contain large numbers of pixels (instances), it can be considered Big Data. In addition, the process of feature selection can be highly parallelized. Thus techniques of big data, e.g., Hadoop,[5] can be used to make it feasible to select features from the full datasets. Finally, we intend to extend our investigation of autoencoder features to other classification problems, e.g., MNIST handwritten digits classification.[6] We are interested in the extent of redundancy and irrelevance of autoencoder features in general.

7. ADDITIONAL AUTHORS

Additional authors: Xiuqin Zhong (University of Electronic Science and Technology of China, email: `zhongxiuqin2009@gmail.com`).

8. REFERENCES

[1] B. Bhanu and Y. Lin. Genetic algorithm based feature selection for target detection in SAR images. *Image and Vision Computing*, 21:591–608, 2003.

[2] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold. Page segmentation of historical document images with convolutional autoencoders. In *13th International Conference on Document Analysis and Recognition*, 2015.

[3] K. Chen, M. Seuret, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold. Ground truth model, tool, and dataset for layout analysis of historical documents. In *Proc. SPIE 9402, Document Recognition and Retrieval XXII*, 2015.

[4] L. Chen, F. Rottensteiner, and C. Heipke. Feature descriptor by convolution and pooling autoencoders. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1:31–38, 2015.

[5] A. Fischer, V. Frinken, A. Fornés, and H. Bunke. Transcription alignment of Latin manuscripts using hidden Markov models. In *Proc. 1st Int. Workshop on Historical Document Imaging and Processing*, pages 29–36, 2011.

[6] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character HMMs. *Pattern Recognition Letters*, 33(7):934–942, 2012.

[7] H. Grabner and H. Bischof. On-line boosting and vision. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 260–267. IEEE, 2006.

[8] M. Gütlein, E. Frank, M. Hall, and A. Karwath. Large-scale attribute selection using wrappers. In *IEEE Symposium on Computational Intelligence and Data Mining*, pages 332–339, 2009.

[9] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[11] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[12] M. Lichman. UCI machine learning repository, 2013.

[13] V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics. *Expert Systems with Applications*, 39(7):6585–6608, June 2012.

[14] J. Lu, T. Zhao, and Y. Zhang. Feature selection based-on genetic algorithm for image annotation. *Knowledge-Based Systems*, 21:887–891, December 2008.

[15] L. Shen and L. Bai. Adaboost Gabor feature selection for classification. In *Proc. of Image and Vision Computing NewZealand*, pages 77–83, 2004.

[16] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12:111–120, 2008.

[17] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using Gabor feature based boosted classifiers. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1692–1698, 2005.

[18] H. Wei, K. Chen, R. Ingold, and M. Liwicki. Hybrid feature selection for historical document layout analysis. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 87–92, 2014.

[19] H. Wei, K. Chen, A. Nicolaou, R. Ingold, and M. Liwicki. Investigation of feature selection for historical document layout analysis. In *4th International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, 2014.

[20] J. Yang and V. Honavar. *Feature Extraction, Construction and Selection*. Springer US, 1998.

[21] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block LBP representation. In *Advances in biometrics*, pages 11–18. Springer, 2007.

[22] B. Zhao, Y.-C. Tam, and J. Zheng. An autoencoder with bilingual sparse features for improved statistical machine translation. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7103–7107. IEEE, 2014.

[23] G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 1453–1461, 2012.

[5] https://hadoop.apache.org/

[6] http://yann.lecun.com/exdb/mnist/