

Article

# Towards Mapping Images to Text Using Deep-Learning Architectures

Daniela Onita <sup>1,2,\*</sup> , Adriana Birlutiu <sup>2,\*</sup> and Liviu P. Dinu <sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Bucharest 90, Panduri Street, Sector 5, 050663 Bucharest, Romania; ldinu@fmi.unibuc.ro

<sup>2</sup> Department of Computer Science and Engineering, “1 Decembrie 1918” University of Alba Iulia 5, Gabriel Bethlen, 515900 Alba Iulia, Romania

\* Correspondence: daniela.onita@uab.ro (D.O.); adriana.birlutiu@uab.ro (A.B.)

Received: 30 June 2020; Accepted: 16 September 2020; Published: 18 September 2020



**Abstract:** Images and text represent types of content that are used together for conveying a message. The process of mapping images to text can provide very useful information and can be included in many applications from the medical domain, applications for blind people, social networking, etc. In this paper, we investigate an approach for mapping images to text using a Kernel Ridge Regression model. We considered two types of features: simple RGB pixel-value features and image features extracted with deep-learning approaches. We investigated several neural network architectures for image feature extraction: VGG16, Inception V3, ResNet50, Xception. The experimental evaluation was performed on three data sets from different domains. The texts associated with images represent objective descriptions for two of the three data sets and subjective descriptions for the other data set. The experimental results show that the more complex deep-learning approaches that were used for feature extraction perform better than simple RGB pixel-value approaches. Moreover, the ResNet50 network architecture performs best in comparison to the other three deep network architectures considered for extracting image features. The model error obtained using the ResNet50 network is less by approx. 0.30 than other neural network architectures. We extracted natural language descriptors of images and we made a comparison between original and generated descriptive words. Furthermore, we investigated if there is a difference in performance between the type of text associated with the images: subjective or objective. The proposed model generated more similar descriptions to the original ones for the data set containing objective descriptions whose vocabulary is simpler, bigger and clearer.

**Keywords:** kernel ridge regression; image captioning; image description; deep learning; convolutional neural network

## 1. Introduction

A quick look at an image is sufficient for a human to say a few words related to that image. However, this very easy task for humans is a very difficult task for existing computer vision systems. The majority of previous work in computer vision [1–4] has focused on labeling images with a fixed set of visual categories. Even though closed vocabularies of visual concepts are a convenient modeling assumption, they are quite restrictive when compared to the vast amount of rich descriptions and impressions that a human can compose.

Some approaches that address the challenge of generating image descriptions have been proposed [5,6]. In this work, we want to take a step forward towards the goal of generating descriptions of the images that are close to the natural language. Figure 1 gives a hint to the motivation of our work by showing several samples that were used in the experimental evaluation. Each sample consists of an image and the text associated with it. We chose three data sets from different domains. The first

data set belongs to the social network domain. The text associated with each image is a subjective description or impression of that image written by a user. The second data set belongs to the medical domain. The text associated with each image is an objective description written by a radiologist. The third data set belongs to the gaming domain and it was formed using a game in which users must label images. The text associated with each image was written by a user and represents descriptive words of images.



**Figure 1.** Motivation Figure: Our model treats language as a rich label space and generates subjective descriptions of images. Examples of samples used in the experimental evaluation. Each sample consists of a pair made of an image and the subjective text associated with it. (A) Samples of Text for Sentiment Analysis (T4SA) data set. (B) Samples of PadChest data set used in the experimental evaluation. The text is written in Spanish language. (C) Samples of ESP Game data set used in the experimental evaluation.

This paper is an extension of our preliminary work which was presented in Recent Advances in Natural Language Processing conference in 2019 [7]. The added contributions of the work described here compared to our preliminary work presented in [7] are:

- Investigating several deep-learning architectures. Based on the preliminary investigations presented in [7], we concluded that the more complex deep-learning approaches are better than the simple RGB pixel values for the feature extraction tasks. For this reason, we further investigated several neural network architectures and we compared the model performance by the complexity of the neural network that was used as feature extractor.
- Experimental evaluation on multiple data sets. We extended the experimental evaluation by using data sets from different domains, i.e., social media, medical data and gaming.
- Qualitative analysis. In addition to the quantitative comparisons which were presented in the preliminary work, we further extended the analysis by also including a qualitative analysis. We compared visually the generated description with the original description of an image.
- Objective vs subjective descriptions. In addition to only subjective descriptions that were analyzed in our previous work, we compared generated descriptions with the original description of an image in the context of the text type (subjective vs. objective).
- Language comparison. We investigated whether the language of the text associated with images influences the performance.

Our core insight is that we can map images to natural text by leveraging the image-text data set in a supervised learning approach in which the image represents the input and the text represents the output. We employed a Kernel Ridge Regression model for the task of mapping images to text. We used two types of features: image and text features. The model generates text which consists of a set of words from a dictionary. We used a bag-of-words model to construct the text features. The image features were extracted with deep-learning approaches in the form of four convolutional neural network (CNN) architectures: 1. VGG16, 2. Inception V3, 3. ResNet50, 4. Xception.

The goal of our work is to compare different types of deep neural network architectures to generate descriptions of images. The four types of deep neural network architectures that we investigated were introduced as the winners of ImageNet challenge (2014–2016) [8]. From VGG14 network, which is the winner of ImageNet challenge 2014, the networks have improvements: the number of layers, the pooling layers, the activation and the loss function, the regularization and the optimization, the reduced number of parameters in relation to number of layers. The main challenge is finding a model that is rich enough to simultaneously reason about contents of images and their representations in natural language domain. Additionally, the model should be free of assumptions about specific templates or categories and instead rely on learning from the training data. The model will go beyond the simple objective description of an image and also give the impression that the image could make upon a certain person. An example of this is shown in the image from bottom-right of Figure 1A in which we do not have a captioning or description of the animal in the image but the subjective impression that the image makes upon the looker.

In the experimental evaluation we investigated three data sets from different domains. We designed a system that automatically associates an image to a set of words from a dictionary. Depending on the data set used, these words are not only descriptors of the content of the image, but also subjective impressions and opinions of the image. Two of the three data sets were written in English, while one data set was written in Spanish. In our experiments, we investigated whether the language of the text associated with the images influences the performance of the model.

There is a difference in performance for the four deep network architectures investigated in the experimental evaluation: the network whose architecture contains the deepest layers has the best results for all three data sets. In particular, the mapping images to text has the best results for the ResNet50 network architecture used as image feature extractor. In terms of similarity, if we compare generated description with the original description of an image, our proposed model performs better for the data set which has objective descriptions associated with images.

The novelty of our contribution is as follows:

- We designed a system that automatically associates an image to a set of words from a dictionary using a Kernel Ridge Regression model. We showed that Kernel Ridge Regression, which is a combination between ridge regression and classification can be used in the problem of image description.
- Based on the experimental evaluation, we confirm the potential of deep-learning techniques for image to text mapping. We considered two types of image features: RGB pixel-value features, and features extracted with four deep-learning approaches. The experimental results show that the features extracted using deep-learning architectures perform better than the RGB pixel-value features. Furthermore, the network whose architecture contains the most hidden layers performs best.
- We investigated the difference between objective and subjective descriptions in three data sets from different domains: social media, medical and gaming domain. We noticed that our model generated more similar words to the original ones for objective descriptions. Furthermore, we noticed that the language of the text associated with images influences the performance of the algorithm.
- The proposed method can predict text that are close to the original text associated with the image. In the experimental evaluation we compared the generated description with the original description of an image: our proposed method performs better for the data set with objective descriptions.
- We consider that investigating different deep-learning architectures for feature extraction for mapping images to text is the added value of our work. To our knowledge, our approach based on a combination of ridge regression and deep learning has not been investigated for mapping images to text before.

## 2. Related Work

*Image to text mapping.* Image to text mapping can be divided into two categories: image captioning and image description. Several approaches for image captioning and image description tasks have been proposed [1,2,4,5]. State-of-the-art techniques for image captioning and image description tasks are based on recurrent neural networks [1,9].

Image captioning can be defined as an automated objective description of an image. This concept congregates two major areas of research: Computer Vision and Natural Language Processing. Organizing words into a sentence is not an easy task for a computer, image captioning needs a high level of understanding of semantic content of an image and the ability to express image information in a human sentence. State-of-the-art techniques for image captioning tasks are based on recurrent neural networks, which take as an example a representation of the characteristics of an image. The research presented in this paper is in the direction of image captioning. Mapping images to text allows us to build some dictionaries of words and select from these dictionaries the words which are the most relevant to an image. The learning setting that we investigate in this paper is different to the image captioning setting because our system automatically associates an image to a set of words from a dictionary, these words being not only descriptors of the content of the image, but also subjective opinions of the image. deep-learning techniques are often used for image captioning tasks [1,2,4]. In [10] the authors proposed a new learning method named Contrastive Learning, which encourages distinctiveness, but at the same time aims to maintain the quality of the generated captions. A reference model was used during the learning process in addition to the image-text pairs. The authors introduced inadequate pairs as input, where the text is the description of another image. In [11] the authors proposed an approach for training an image captioning model in an unsupervised manner. In contrast to our setting their model requires an image set, a sentence corpus, and an existing visual concept detector.

Image description is more than image captioning. In image captioning tasks, the text associated with an image represents an objective description, while in image description tasks the text associated with an image is a subjective description. Several approaches that address the challenge of generating image descriptions have been proposed [5,6,9,12]. In [9] a hierarchical Recurrent Neural Network model based on the phrase was presented, which incorporates the natural language provided by the human expert. We considered that the task of generating objective description is in the image captioning domain, while the task of generating a subjective description is in the domain of image description. We used three data sets from different domains for experimental evaluation. Two data sets contain text in the form of objective descriptions and one data set contains subjective descriptions. The text from the latter data set is a subjective description of the image written by a user. In [5], the authors developed a multimodal Recurrent Neural Network architecture that is capable of generating a description for an input image. The model can find visual-semantic connections, even if the image shows a small object. In contrast with the model presented in [5], our approach associates an image to a set of words from a dictionary. The dictionary is formed based on the image descriptions which were provided by the users. In comparison with [5], our proposed model can also generate subjective descriptions.

**Multimodal High-level Representation using Deep Learning.** Multimodal data refers to the multiple modalities/types of information that are used in a research problem or an experience. Multimodal deep neural networks have been very successful in computer vision and natural language applications [13–15]. In [16] the authors investigated multimodal learning using audio and video data. The authors of [14] presented an approach to learn several specialist models using deep-learning techniques using video and audio information. In our study, we combined two types of information for improving the mapping images to text problem: images and associated tags or text explanations. Based on the information type used in a research, different types of information representations was developed for feature extraction. For visual information type, CNN are the main approach for image high-level representation [4,17]. We compared four deep neural network architectures for images high-level representation in the context of mapping images to text. For textual features, the authors of [18] proposed a document ranking model composed of two separate deep neural networks, one that matches the query and the document using a local representation, and another that matches the query and the document using learned distributed representations. In our model, for textual data representation, we used a simplifying representation which represents text as a multiset.

**Kernel Ridge Regression.** Kernel Ridge Regression (KRR) [19] combines Ridge Regression and classification using the kernel trick. KRR techniques have been applied for solving various problems, such as face recognition [20], multi-class classification [21], genome selection [22], pattern prediction [23].

### 3. Materials and Methods

#### 3.1. Kernel Ridge Regression for Mapping Images to Text

In this section, we describe the Kernel Ridge Regression model that we use for mapping images to text.

Let  $X = \{x_1, x_2, \dots, x_n\}$  and  $Y = \{y_1, y_2, \dots, y_n\}$  be the set of inputs and outputs, respectively, and  $n$  represents the number of observations. And let  $F_X \in \mathbb{R}^{d_X \times n}$  and  $F_Y \in \mathbb{R}^{d_Y \times n}$  denote the input and output feature matrices, where  $d_X, d_Y$  represent the dimensions of the input and output features respectively. The inputs represent the images, and the input features can be either simple RGB pixel values or something more complex, such as features extracted automatically using convolutional neural networks [24]. The outputs represent the texts associated with the images and the output features can be extracted using Word2Vec [25].

A mapping between the inputs and the outputs can be formulated as a multi-linear regression problem [26,27]. Combined with Tikhonov regularization, this is also known as Kernel Ridge Regression (KRR). The KRR method is a regularized least squares method that is used for classification and regression tasks. It has the following objective function:

$$\arg_W \min(\frac{1}{2} \|WF_X - F_Y^T\|_{\mathcal{F}}^2 + \alpha \frac{1}{2} \|W\|_{\mathcal{F}}^2) \tag{1}$$

where  $\|\cdot\|_{\mathcal{F}}$  is the Frobenius norm,  $\alpha$  is a regularization term and the superscript  $T$  signifies the transpose of the matrix. The solution of the optimization problem from Equation (1) involves the Moore-Penrose pseudo-inverse [28] and has the following closed-form expression:

$$W = F_Y F_X^T (F_X F_X^T + \alpha I_{d_X})^{-1} \in \mathbb{R}^{d_Y \times d_X} \tag{2}$$

which for low-dimensional feature spaces ( $d_X, d_Y \leq n$ ) can be calculated explicitly (the  $I_{d_X}$  in Equation (2) represents the identity matrix of dimension  $d_X$ ). For high-dimensional data, as in the case for image data, an explicit computation of  $W$  as presented in Equation (2) without prior dimensionality reduction is computationally expensive. Fortunately, the closed-form solution can be computed via inversion of the Gram matrix of  $F_X$  instead of the covariance matrix, given the following relation [28]:

$$(P^{-1} + X^T R^{-1} X)^{-1} X^T R^{-1} = P X^T (X P X^T + R)^{-1} \tag{3}$$

We substitute  $X = F_X$ ,  $R = \alpha I_{d_X}$ , and  $P = I_n$  where  $I_{d_X}$ ,  $I_n$  are the  $d_X$ - and  $n$ -dimensional identity matrices, respectively. Hence, Equation (2) can be rewritten to:

$$\begin{aligned} W &= F_Y F_X^T (F_X F_X^T + \alpha I_{d_X})^{-1} \\ &= F_Y (F_X^T F_X + \alpha I_n)^{-1} F_X^T \end{aligned} \tag{4}$$

Even further, Equation (4) can be augmented by applying the kernel trick. The inputs  $x_i$  are implicitly mapped to  $\phi(x_i)$  in a high-dimensional Hilbert space [29]:

$$\Phi = [\phi(x_1), \dots, \phi(x_n)]. \tag{5}$$

when predicting a target  $y_{new}$  from a new observation  $x_{new}$ , explicit access to  $\Phi$  is never actually needed:

$$\begin{aligned} y_{new} &= F_Y (\Phi^T \Phi + \alpha I_n)^{-1} \Phi^T \phi(x_{new}) \\ &= F_Y (K + \alpha I_n)^{-1} \kappa(x_{new}) \end{aligned} \tag{6}$$

With  $K_{ij} = \phi(x_i)^T \phi(x_j)$  and  $\kappa(x_{new})_i = \phi(x_i)^T \phi(x_{new})$ , the prediction can be described entirely in terms of inner products in the higher-dimensional space. Not only does this approach work on the original data sets without the need for dimensionality reduction, but it also opens up ways to introduce non-linear mappings into the regression by considering different types of kernels, such as Gaussian or polynomial kernels.

The schematic representation of the proposed framework for mapping images to text is shown in Figure 2. The image and text features are used as input and output features for the proposed KRR model.

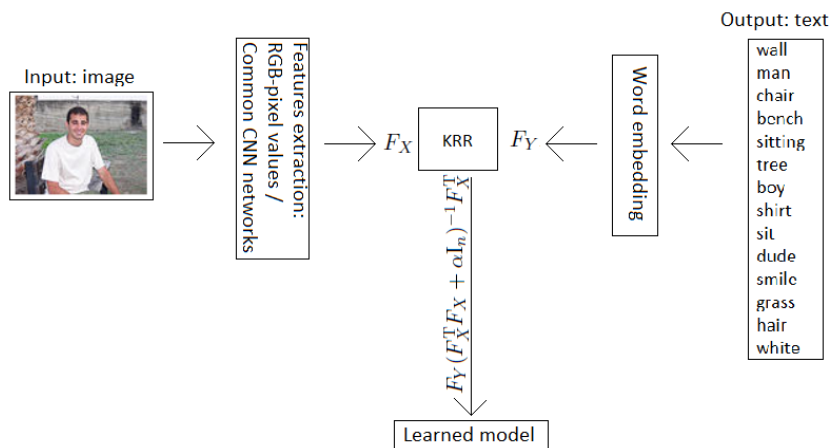


Figure 2. Framework of the proposed model.

### 3.2. Data Sets

We evaluated the proposed KRR model for mapping images to text on three data sets from different domains. The data sets contain images and text associated with each image. For one data set the text is a description or impression of the image, written by a user. We also used for experimental evaluation a data set in which the text associated with the images is a radiologist’s report written in Spanish language. For the third data set the text associated with images is a set of descriptive words, written by a user. We describe these data sets in the following.

Text for Sentiment Analysis (T4SA). This data set was introduced in [30]. The data have been collected from Twitter posts over 6 months, and using an LSTM-SVM (Long-Short Term Memory—Support Vector Machine) architecture, the tweets have been divided into three sentiment categories: positive, neutral, and negative. For image labeling, the authors have selected the data with the most confident textual sentiment predictions, and they used these predictions to automatically assign sentiment labels to the corresponding images. In our experimental evaluation we selected 10 k images and the corresponding 10k tweets from each of the three sentiment categories. Figure 1A shows examples of images and the associated texts from this data set.

PADchest data set. PadChest data set [31] is a public corpus which was collected in Spain at Hospital San Juan from 2009 to 2017. It includes more than 160k X-rays images. The X-rays images were interpreted by radiologists and each image was associated with a report written in Spanish language. 27% from reports were manually annotated by trained physicians and the remaining set was labeled using a supervised method based on a recurrent neural network with attention mechanisms. In our experimental evaluation we selected 10k images and the corresponding 10k reports. Figure 1B shows samples from this data set.

ESP Game data set. The ESP Game data set [3] is a public data set from Kaggle. The ESP Game is an online game that awarded players points if they could label an image with the same word as another unknown player logged in from a different location. The ESP Game data set consists of 100 k images and each image has a list of words associated with it. The data set was labeled using ESP Game. We selected 10 k images and corresponding 10k descriptions for the experimental evaluation. Figure 1C shows samples of the ESP Game data set.

### 3.3. Image Features

The research on feature extraction from images proceeds along two directions: (i) traditional, hand-crafted features, and (ii) automatically generated features. With the increasing number of images and videos on the web, traditional methods have a hard time handling the scalability and generalization problem. In contrast, automated generated feature-based techniques are capable of automatically learning robust features from a large number of images [32].

To emphasize the advantage of deep-learning techniques for image high-level representation, we compared the performance of four CNN architectures used for image representation with the process of the simply converting of the images into arrays for extracting features from images. Each image was sliced to get the RGB data. The 3-channels RGB image format was preferred instead of using 1-channel image format since we wanted to use all the available information related to an image. Using this approach, each image was described by a 2352 ( $28 \times 28 \times 3$ )-dimensional feature vector.

Deep-learning models use a cascade of layers to discover feature representations from data. Each layer of a convolutional network produces an activation for the given input. Earlier layers capture low-level features of the image like blobs, edges, and colors. These primitive features are abstracted by the high-level layers. Studies from the literature suggest that while using pre-trained networks for feature extraction, the features should be extracted from the layer right before the classification layer [17]. For this reason, we extracted the features from the last layer before the final classification, so the entire convolutional base was used for this.

For understanding the features of an input image and how the networks work, it is important to understand how convolution and pooling layers are calculated. Convolutional parameters can be used for reducing some features in the image which can be ignored in the training process. The following hyperparameters are used for calculating the number of network parameters: number of filters ( $k$ ), filter width ( $Fw$ ), filter height ( $Fh$ ), stride width ( $Sw$ ), stride height ( $Sh$ ) and padding ( $P$ ). To determinate the receptive field (the size of the region in the input that produces the feature) described by output width ( $Ow$ ) and output height ( $Oh$ ), the following equations are used:

$$Ow = \left( \frac{W - Fw + 2P}{Sw} \right) + 1 \quad (7)$$

$$Oh = \left( \frac{H - Fh + 2P}{Sh} \right) + 1 \quad (8)$$

The following formula is used to calculate the pooling layer:

$$\left( \frac{IM + 2P - F}{S} + 1 \right) \quad (9)$$

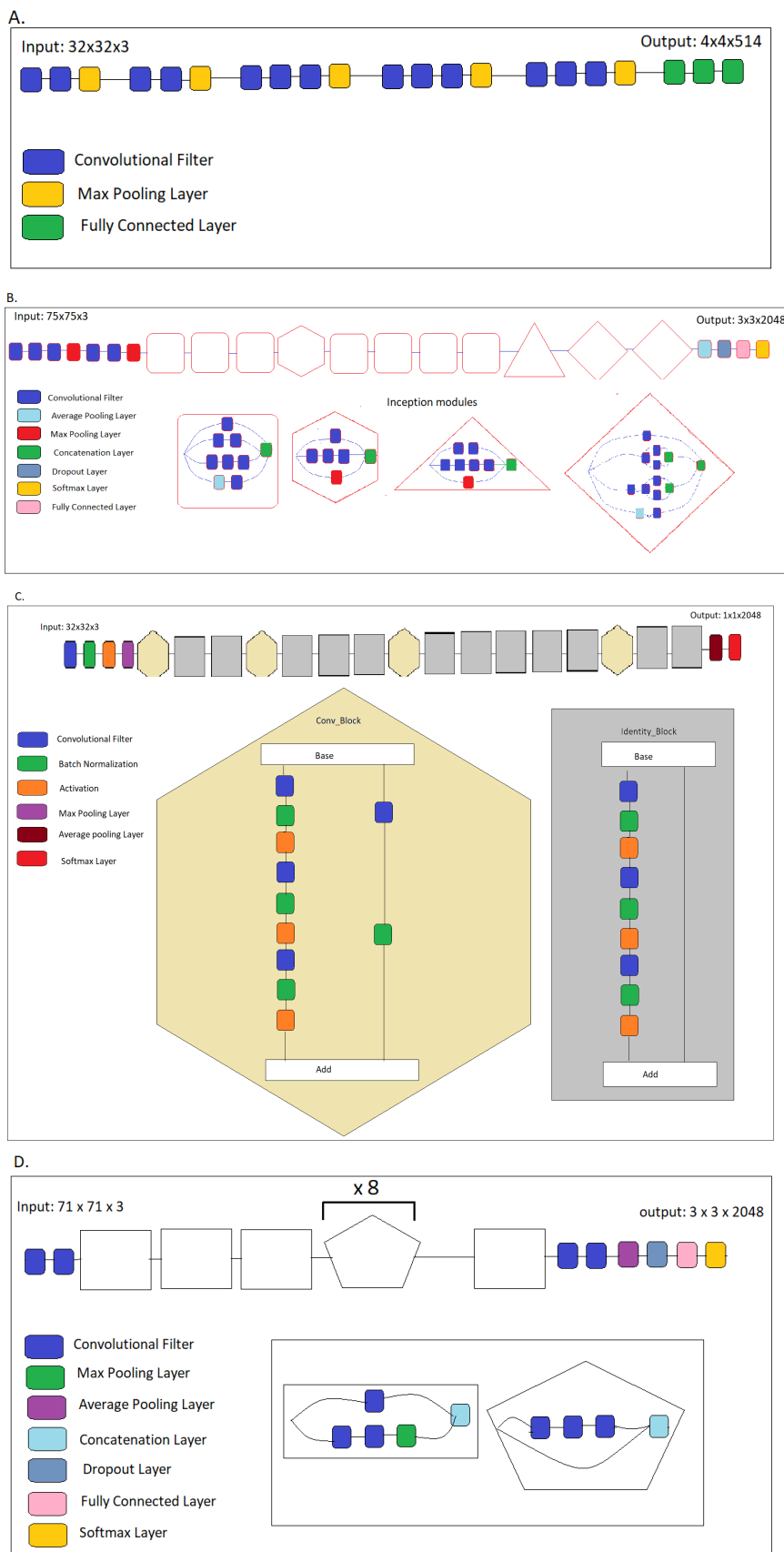
where IM is the input matrix, F is the filter and S represents the stride. Starting from the input image and applying the above formulas, the convolutions, pooling and feature map outputs will be obtained.

We investigated four different network architectures: VGG16, Inception V3, ResNet50 and Xception. The four architectures differ in the number of layers and the number of parameters. These network architectures were chosen because they are the most popular CNN architectures. Each network has different improvements to the first CNN architecture (AlexNet) which was developed in 2012 [33].

### 3.3.1. Vgg16

The VGG16 network architecture was introduced in 2014 [34]. VGG16 brings several improvements over AlexNet: fewer parameters, a large number of weight layers, the decision function is more discriminative, to name just a few. The large kernel sized filters from first and the second convolutional layer from AlexNet architectures were replaced with multiple  $3 \times 3$  kernel sized filters in the VGG16 architecture. The VGG16 has a uniform architecture with 16 hidden layers and 138 million of trainable parameters. For computational reasons, while the features were extracted using VGG16 architecture the images were resized to a 3072-pixel resolution. The VGG16 was initialized by the ImageNet weights. Figure 3A shows the graphical representation of the VGG16 network.





**Figure 3.** The graphical representation of the four network architectures investigated for extracting deep-learning features from images. **(A)** VGG16 architecture. **(B)** Inception V3 architecture. **(C)** ResNet50 architecture. **(D)** Xception architecture.

### 3.3.2. Inception

The Inception network was introduced in 2014 (Inception V1 [35]) as the winner of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [8]. The architecture of this type of network consists of 22 hidden layers and a reduced number of parameters to 7 million (for comparison, AlexNet, the first CNN architecture, has 60 million parameters). Inception networks contain a module called inception module which approximates a sparse CNN with a normal dense construction. Inception V3 network was introduced in [36] and it has 42 hidden layers and 23 million parameters. The architecture of Inception V3 consists of 11 inception modules. Each module is formed by pooling layers, convolutional filters and an activation function which in this case is the rectified linear unit function. Although the image features were extracted using Inception V3 architecture, the input size of the images was resized to  $75 \times 75$  and a-dimensional feature vector of size  $3 \times 3 \times 2048$  from the final convolutional layer was returned. Figure 3B shows the graphical representation of the Inception V3 network architecture.

### 3.3.3. Residual Networks

Residual Networks (ResNets) [37] are a type of classical neural networks that were introduced in 2015 as the winner model of ImageNet challenge (ILSVRC 2015) [8]. A network has a Residual Network architecture if in addition to the convolution, pooling, activation and fully connected layers it has also the identity connection between the layers. A residual block can be represented mathematically as follows: [38]

$$y = F(x, W_i) + x \quad (10)$$

where  $y$  is the output function,  $x$  is the input to the residual block.  $W_i$  represents the weight layers contained by residual block, where  $1 \leq i \leq$  number of layers in a residual block. If the residual block contains 2 weight layers, the residual block  $F(x, W_i)$  can be written as follows:

$$F(x, W_i) = W_2\sigma(W_1x) \quad (11)$$

where  $\sigma$  is the ReLU activation function and  $\sigma$  is calculated using the equation:

$$\sigma = \max(0, x) \quad (12)$$

In comparison to VGG networks, the evaluation time was reduced when the residual networks are used. ResNet50 is a convolutional network with a deep of 50 hidden layers and over 23 million trainable parameters. The network requires an image of input size  $224 \times 224$  pixels and 3 input channels, but this size can be lower for computational reasons. In the experimental evaluation, we used an input image of size  $32 \times 32 \times 3$  and the ResNets50 returned a 2048-dimensional feature vector. Figure 3C shows the graphical representation of the ResNets architecture.

### 3.3.4. Xception

The Xception network was introduced in 2016 [39]. The Xception architecture is an improved version of the Inception V3 architecture: the inception modules have been replaced with depthwise separable convolutions. The architecture of Xception network consists of 36 convolutional layers that form the feature extraction base. The Xception network is structured into 14 modules. Except for the first and last modules, each convolutional layer has residual connections around them. The regular input size of images is  $224 \times 224$ , but from computational reasons, we used the smallest possible size,  $71 \times 71$ . The output of the convolutional base has a size of  $3 \times 3 \times 2048$ . Figure 3D shows the graphical representation of the Xception network.

### 3.3.5. Comparison of Network Architectures

For comparison purposes, we designed the graphical representations of the four deep-learning architectures which are shown in the Figure 3. The graphical representations help us to visualize the similarities and differences between the four architectures and to get insights related to them. Although the VGG16 architecture is formed by convolutional, max pooling and fully connected layers, the other three architectures are built by modules and blocks of layers. Instead of stacking convolutional layers, Inception V3, ResNet50 and Xception networks stack modules or blocks, within which are convolutional layers. It is obvious from the graphics of Figure 3 that the architecture of Inception V3 network was improved in comparison to the first two architectures: the types of the layers, the inception modules, the number of layers. The graphical representations indicate that ResNet50 network used batch normalization and the skip connection concept. One can notice that from VGG16 network, which was developed in 2014, network architectures became more complex and they have different improvements to AlexNet—the first CNN architecture: number of layers, using modules, number of pooling layers, activation and loss function, regularization and optimization. One can also notice that even if the number of layers increases, the number of parameters decreases.

Table 1 presents the comparison between the network architectures described above, regarding the number of layers and the number of parameters.

**Table 1.** Comparison of the four network architectures considered.

Year	CNN	Developed By	No. of Hidden Layers	No. of Parameters
2014	VGG 16 Net [34]	Simonyan Zisserman	16	138 million
2015	Inception V3 [36]	Google	42	23 million
2015	ResNet50 [37]	Kaiming He	50	26 million
2016	Xception [39]	Francois Chollet	48	23 million

### 3.4. Text Features

We used a Bag-of-Words (BoW) model [40] for extracting the features from the text samples. The first step in building the BoW model consists of pre-processing the text: removing non-letter characters, removing the html tags, converting words to lower cases, removing stop-words and making the split. Vocabulary is built from the words that appear in the text samples. The input of the BoW model is a list of strings and the output is a sparse matrix with the dimension: number of samples  $\times$  number of words in the vocabulary, with 1 if a given word from the vocabulary is contained in that particular text sample. We initialized the BoW model with a maximum of 5000 features. We extracted vocabulary for each data set, and the corresponding 0-1 feature vector for each text sample.

## 4. Experimental Protocol

We designed an experimental protocol that would help us answer the following questions:

1. Could our proposed Kernel Ridge Regression model map images to natural language descriptors?
2. What is the difference between the four types of network architectures that we considered? Also, we are interested in whether the more complex deep-learning features give a better performance in comparison to the simple RGB pixel-value features. Which of the four deep-learning network architectures performs best? Can we draw some insights from this comparison?

To answer these questions, we designed the following experimental protocol. For each of the three data sets, we randomly split the data 5 times into training and test set, taking 70% from the data set for training and the rest for testing. For training the model, we considered different sizes of the training set: from 50 to 7000 observations with a step size of 50. For a correct evaluation, the models built on these different training sets were evaluated on the same test set. The error was averaged over the 5 random splits of the data into training and test set.

### *Evaluation Measure*

To measure how good our models map images to text, we developed a specific evaluation measure. The reason was that each output of our model represents a very large vector of probabilities, with the dimension equal to the number of words in the dictionary (approximately 5000 components). Each component of the output vector represents the probability of the corresponding word from the vocabulary as being a descriptor of that image. Given this particular form of the output, the evaluation measure was computed as follows: 1. we sorted in descending order the absolute values of the predicted output vector; 2. we created a new vector containing the first 50 words from the predicted output vector; 3. we computed the Euclidean distance between the predicted output vector values and the actual output vector.

The actual output vector is a sparse vector, a component in this vector is 1 if the corresponding word from the vocabulary is contained in that particular description of the image. The values computed in the third step described above were averaged over the entire test data set and the average value obtained was considered to be the error.

## **5. Results and Discussions**

### *5.1. Quantitative Analysis*

The first questions raised above can be answered by analyzing the experimental results shown in Figure 4. The plots show the learning curve (mean errors and standard deviations) for different sizes of the training set and different sentiment categories. Since the error decreases as the training size increases, it is obvious that there is learning involved, thus our proposed model can map images to natural language descriptors. The plots from Figure 4 also show the comparison between the RGB pixel values and VGG16 features for the three sentiment categories considered. Overall, the more complex deep-learning features give a better performance in comparison to the simple RGB pixel-values features.

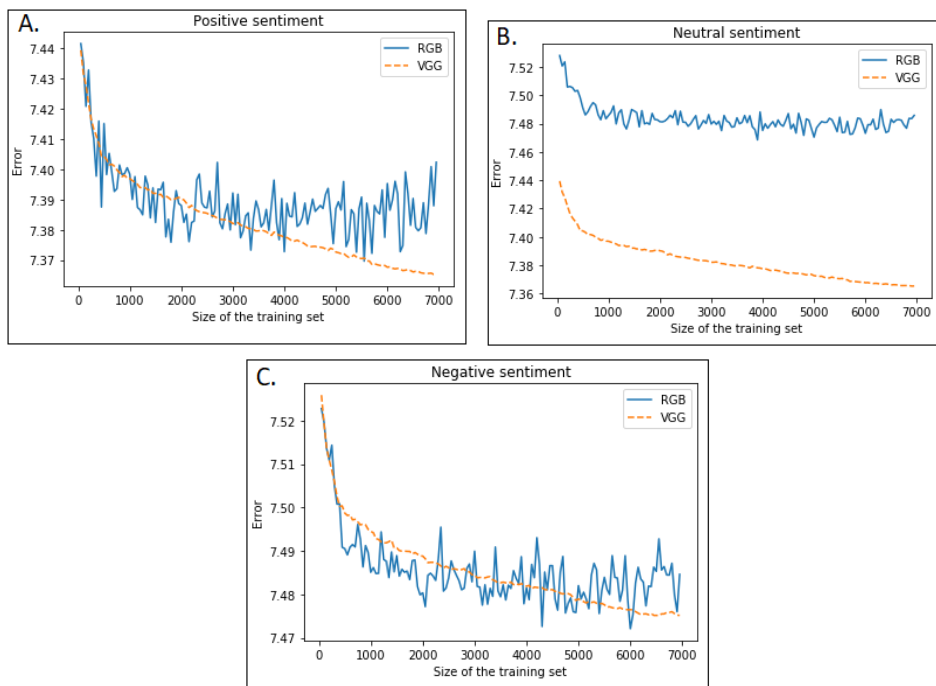
Furthermore, we can also see from Figure 4 that the neutral sentiment category has different behavior in comparison with the positive and negative sentiment categories. In the case of neutral sentiment, the more complex VGG16 features have a better performance than the simpler RGB pixel-value features as the size of the data increases. For positive and negative sentiment categories, the simpler RGB pixel-value features lead to an error which varies a lot, while using the VGG16 features, the error is more stable.

The second question can be answered by analyzing the experimental results shown in Figure 5. The plots from Figure 5A show mean errors and standard deviation for different sizes of the training set from the T4SA data set. The more complex ResNet50 features have a better performance than the simpler RGB pixel-value features and than the other three CNN architectures.

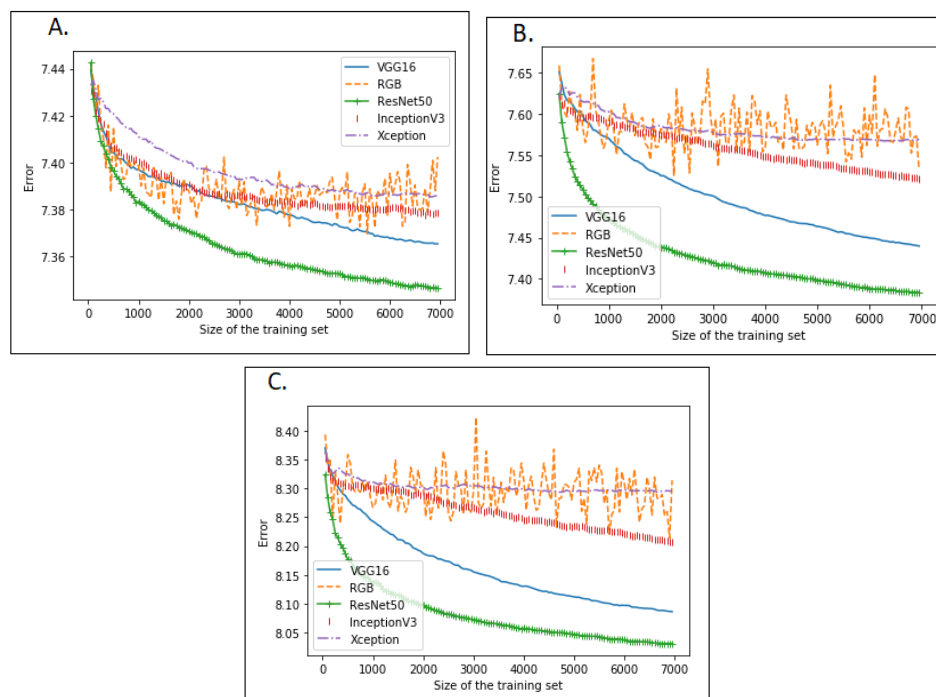
The plots from Figure 5B show the learning curve (mean errors and standard deviations) for different sizes of the training set for ESP Game data set using different deep-learning techniques for image extraction. The KRR model performs best when the ResNet50 network is used as image feature extractor, but essentially the more complex deep-learning features give a better performance in comparison to the simple RGB pixel-values-features.

Figure 5C shows the comparison between RGB pixel-value features, VGG16 features, ResNet50 features, InceptionV3 features and Xception features for PadChest data set. The proposed model performs better when image features are extracted using deep-learning techniques.

Figure 5 shows that our model with ResNet50 as image feature extractor performs best for three data sets. Overall, the more complex deep-learning features give a better performance in comparison to the simple RGB pixel-value features.



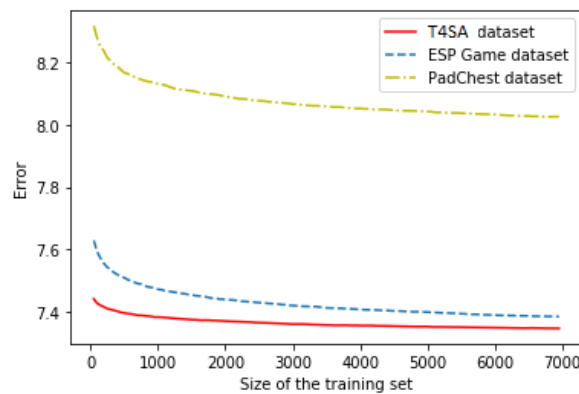
**Figure 4.** Mean errors and standard deviation for different sizes of the training set for T4SA data set. Comparison between RGB pixel-value features and the VGG16 features. (A)—positive sentiment category, (B)—neutral sentiment category, (C)—negative sentiment category.



**Figure 5.** Comparison between RGB pixel-value features, VGG16 features, ResNet50 features, InceptionV3 features and Xception features. (A) Text for Sentiment Analysis data set. (B) ESP Game data set. (C) PadChest data set.

The plots from Figure 6 show the comparison of the learning curve for the three data set using ResNet50 as image feature extractor. There is a close value of the error for the ESP Game data set and


T4SA data set. The error for PadChest data set whose text is written in the Spanish language is higher compared to the other two data sets whose description is written in the English language.



**Figure 6.** Comparison of the learning performance for the three data sets using the ResNet50 image features.

5.2. Qualitative Analysis

To answer the first question from Section 4, we analyzed in more detail the natural language descriptors returned by our proposed KRR model. Figure 7 shows the natural language descriptors returned by our model using the four types of image features that we considered. The mapped image from Figure 7 is from the ESP Game data set. We compared the description returned by our model with the original image description. When our model uses image features extracted with the Xception network, only three words by 20 correspond to the original image description. When the model uses as image feature extraction VGG16, ResNet50 and InceptionV3 networks, five similar words with the original description were returned. However, if we look at the returned words, we can identify many correct image descriptors. For example, using InceptionV3 as image feature extraction, the followings words describe the image: “eyes”, “face”, “gray”, “hands”, “person”.

	<b>Original description:</b>					
	wall					
	man					
	chair					
	bench					
	sitting					
	tree					
	boy					
	shirt					
	sit					
	dude					
	smile					
	grass					
	hair					
	white					
		<b>VGG16:</b>		<b>ResNet50:</b>	<b>InceptionV3</b>	<b>Xception</b>
		24 ad	24 ad	759 chair	240 back	
		403 black	403 black	1519 eyes	551 brown	
		432 blue	432 blue	1522 face	577 building	
		551 brown	551 brown	1815 girl	601 bush	
	1522 face	1519 eyes	1888 gray	1724 french		
	1821 glasses	1815 girl	1925 hair	1820 glass		
	1894 green	1894 green	1946 hands	1881 grass		
	1925 hair	1897 grey	2605 man	1894 green		
	2605 man	1925 hair	3192 person	1925 hair		
	2996 old	2479 light	3551 red	1943 handle		
	3178 people	2538 logo	3894 shirt	2056 home		
	3222 picture	2605 man	4256 street	2094 house		
	3551 red	2945 nose	4297 suit	2358 knob		
	3985 sky	3551 red	4594 trees	2978 ocean		
	4018 smile	3937 sign	4814 water	3048 outside		
	4593 tree	3985 sky	4868 white	3551 red		
	4814 water	4018 smile	4891 window	3935 siding		
	4868 white	4593 tree	4912 woman	3985 sky		
	4912 woman	4868 white	4913 women	4868 white		
	4913 women	4912 woman	4917 wood	4891 window		

**Figure 7.** ESP Game Data set: Comparison between original and generated description using different types of image features. Similar descriptors are marked in red rectangle.

To generate the descriptors from Figure 7, we considered 7000 observations as the size of the training set. If we considered the size of the training set from 50 to 70 observations with a step size of 50, three words were similar to the original description using ResNet50, InceptionV3 and Xception networks and only two similar words with the original description when VGG16 network was used as image features extractor.

We initialized the BoW model with a maximum of 5000 features. The extracted vocabulary for the ESP Game data set has 5000 words.

Figure 8 shows the natural language descriptors generated using our KRR model and the original description of an image from the PadChest dataset. To generate the descriptive words we considered different sizes of the training set: from 50 to 7000 observations with a step size of 50 and an extracted vocabulary containing 3623 words. The vocabulary size is smaller due to the repetition of words in the physician report and this fact may affect the performance of the model for generating similar words.

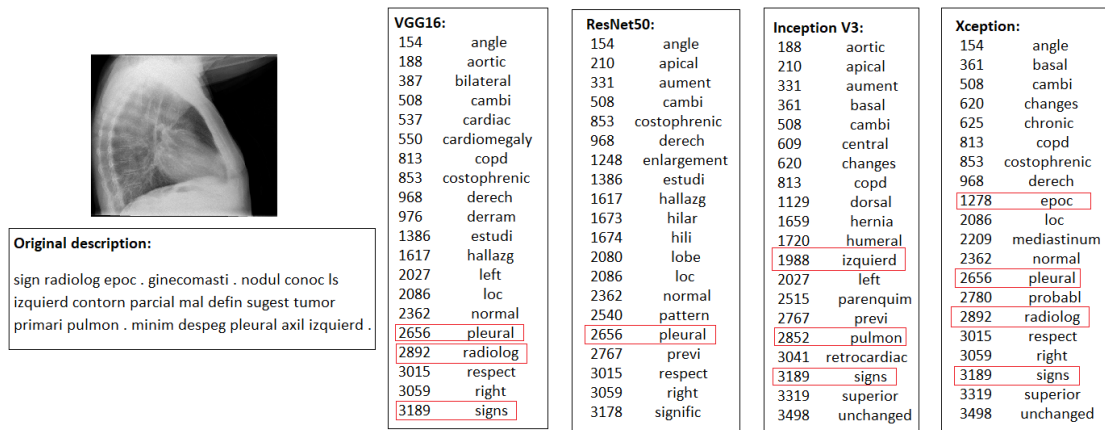


Figure 8. PadChest Data set: Comparison between original and generated description using different types of image features. Similar descriptors are marked in red rectangle.

Figure 9 shows the image description for an image from T4SA data set. The proposed model performs better when image features are extracted with ResNet50 network architecture. In that case, our model returned 3 similar words to the original description.

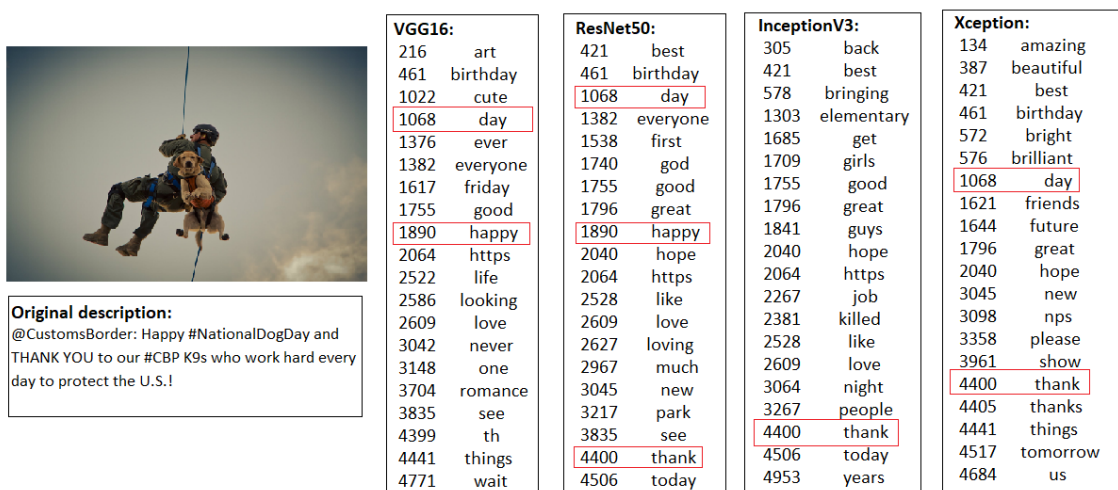


Figure 9. Text for Sentiment Analysis data set: Comparison between original and generated description using different types of image features. Similar descriptors are marked in red rectangle.

The qualitative analysis revealed that our proposed KRR model proposed the KRR model performs better on the ESP Game data set, which contains objective descriptions, in comparison to the PadChest and T4SA data sets.

The experiments show that the model performance is better for the data sets whose text are in English. This can be seen in Figure 6 which shows the comparison of the learning performance for the three data sets using ResNet50 network architecture as image feature extractor. For the PadChest data set, whose texts associated with images were written in Spanish, the model returned the largest error in comparison with the other two data sets whose texts were written in English.

## 6. Conclusions

In this work, we investigated a method for mapping images to text in different real-world scenarios. The mapping from images to text was performed using a Kernel Ridge Regression model. Several deep-learning approaches were used for image descriptor calculation, including VGG16, Inception V3, ResNet50, and Xception. To confirm the potential of deep-learning techniques for mapping images to text, we considered two types of features: simple RGB pixel-value features and image features extracted with deep-learning approaches. The experimental evaluation showed that the features extracted using different CNN architectures perform better than the RGB pixel-value features. We found that there is a difference in performance for different data sets and different deep-learning architectures, in particular the mapping performs better using ResNet50 as image feature extractor, which has the largest number of hidden layers compared to the other three networks considered for the experimental evaluation. The results show that the model error obtained using the ResNet50 architecture is less by approx. 0.30 than the errors obtained with the other neural network architectures considered.

The experimental evaluation was performed on three data sets from different domains, each data set containing both text and images. We made a comparison between the original text and the generated text by our proposed model. The results showed that the proposed method can predict text that is close to the original one. We investigated the difference between objective and subjective text descriptions of images. Our method generated words more similar to the original descriptions of images for the data set whose text consists of objective descriptors associated with images.

As future work, we plan to further extend our approach by investigating the multimodal machine translation process [41] and to integrate into our model textual captions of images obtained using a pre-trained network [34]. The textual captions could be used as a new type of feature and can be compared and integrated with the other image features considered.

**Author Contributions:** Conceptualization: D.O., L.P.D. and A.B.; Methodology: A.B.; Software: D.O.; Validation: D.O. and A.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
RGB	Red Green Blue
KRR	Kernel Ridge Regression
NLP	Natural Language Processing
BoW	Bag-of-Words

## References

1. Bai, S.; An, S. A survey on automatic image caption generation. *Neurocomputing* **2018**, *311*, 291–304. [[CrossRef](#)]



2. Singam, P. Automated Image Captioning Using ConvNets and Recurrent Neural Network. *Int. J. Res. Appl. Sci. Eng.* **2018**, *6*, 1168–1172. [[CrossRef](#)]
3. Von Ahn, L.; Dabbish, L. Labeling images with a computer game. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, 24–29 April 2004; pp. 319–326.
4. Hossain, M.Z.; Sohel, F.; Shiratuddin, M.F.; Laga, H. A comprehensive survey of deep learning for image captioning. *Acm Comput. Surv. (CSUR)* **2019**, *51*, 1–36. [[CrossRef](#)]
5. Karpathy, A.; Li, F.-F. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3128–3137.
6. Kulkarni, G.; Premraj, V.; Ordonez, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, T.L. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2891–2903. [[CrossRef](#)] [[PubMed](#)]
7. Onita, D.; Dinu, L.P.; Birlutiu, A. From Image to Text in Sentiment Analysis via Regression and Deep Learning. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 2–4 September 2019; pp. 862–868.
8. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
9. Fidler, S. Teaching Machines to Describe Images with Natural Language Feedback. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5068–5078.
10. Bo, D.; Duhua, L. Contrastive Learning for Image Captioning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 898–907.
11. Feng, Y.; Ma, L.; Liu, W.; Luo, J. Unsupervised image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4125–4134.
12. Park, C.; Kim, B.; Kim, G. Attend to you: Personalized image captioning with context sequence memory networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 895–903.
13. Goh, G.B.; Sakloth, K.; Siegel, C.; Vishnu, A.; Pfaendtner, J. Multimodal Deep Neural Networks using Both Engineered and Learned Representations for Biodegradability Prediction. *arXiv* **2018**, arXiv:1808.04456.
14. Kahou, S.E.; Bouthillier, X.; Lamblin, P.; Gulcehre, C.; Michalski, V.; Konda, K.; Jean, S.; Froumenty, P.; Dauphin, Y.; Boulanger-Lewandowski, N.; et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *J. Multimodal User Interfaces* **2016**, *10*, 99–111. [[CrossRef](#)]
15. Xu, T.; Zhang, H.; Huang, X.; Zhang, S.; Metaxas, D.N. Multimodal deep learning for cervical dysplasia diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2016; pp. 115–123.
16. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning (ICML'11)*; Omnipress: Madison, WI, USA, 2011; pp. 689–696.
17. Rajaraman, S.; Antani, S.K.; Poostchi, M.; Silamut, K.; Hossain, M.A.; Maude, R.J.; Thoma, G.R. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ* **2018**, *6*, e4568. [[CrossRef](#)] [[PubMed](#)]
18. Mitra, B.; Diaz, F.; Craswell, N. Learning to match using local and distributed representations of text for web search. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1291–1299.
19. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2012.
20. An, S.; Liu, W.; Venkatesh, S. Face recognition using kernel ridge regression. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–7.
21. Rakesh, K.; Suganthan, P.N. An ensemble of kernel ridge regression for multi-class classification. *Procedia Comput. Sci.* **2017**, *108*, 375–383. [[CrossRef](#)]
22. Endelman, J.B. Ridge regression and other kernels for genomic selection with R package rrBLUP. *Plant Genome* **2011**, *4*, 250–255. [[CrossRef](#)]
23. Chu, C.; Ni, Y.; Tan, G.; Saunders, C.J.; Ashburner, J. Kernel regression for fMRI pattern prediction. *NeuroImage* **2011**, *56*, 662–673. [[CrossRef](#)]

24. O’Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
25. Ma, L.; Zhang, Y. Using Word2Vec to process big text data. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2895–2897.
26. Cortes, C.; Mohri, M.; Weston, J. A general regression technique for learning transductions. In Proceedings of the 22nd international conference on Machine learning, Bonn, Germany, 7–11 August 2005; pp. 153–160.
27. Cortes, C.; Mohri, M.; Weston, J. A general regression framework for learning string-to-string mappings. *Predict. Struct. Data* **2007**, *2*. [[CrossRef](#)]
28. Albert, A. *Regression and the Moore-Penrose Pseudoinverse*; Technical Report; Academic Press: New York, NY, USA, 1972.
29. Berlinet, A.; Thomas-Agnan, C. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*; Springer Science Business Media: Berlin, Germany, 2011.
30. Vadicamo, L.; Carrara, F.; Cimino, A.; Cresci, S.; Dell’Orletta, F.; Falchi, F.; Tesconi, M. Cross-media learning for image sentiment analysis in the wild. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 308–317.
31. Bustos, A.; Pertusa, A.; Salinas, J.M.; de la Iglesia-Vayá, M. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *arXiv* **2019**, arXiv:1901.07441.
32. Jindal, S.; Singh, S. Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning. In Proceedings of the 2015 International Conference on Information Processing (ICIP), Pune, India, 16–19 December 2015; p. 447.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing Systems, Stateline, NV, USA, 3–8 December 2012; pp. 1097–1105.
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 July 2015; pp. 1–9.
36. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 630–645.
39. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
40. Harris, Z. Distributional structure. *World* **1954**, *10*, 146–162.
41. Caglayan, O.; Madhyastha, P.; Specia, L.; Barrault, L. Probing the need for visual context in multimodal machine translation. *arXiv* **2019**, arXiv:1903.08678.

