Stat
The ISI's Journal for the Rapid
Dissemination of Statistics Research

(wileyonlinelibrary.com) DOI: 10.100X/sta.0000

MCNN: Masked Convolutional Neural Network for Supervised Learning Problems

Leo Yu-Feng Liu[a], Yufeng Liu[b], Hongtu Zhu[c]

Received XX XX 2020; Accepted XX XX 2020

Convolutional neural networks (CNNs) have exhibited superior performance in various types of classification and prediction tasks, but their interpretability remains to be low despite years of research effort. It is crucial to improve the ability of existing models to interpret deep neural networks from both theoretical and practical perspectives and to develop new neural network models with interpretable representations. The aim of this paper is to propose a set of novel masked convolutional neural network (MCNN) models with better ability to interpret networks and more accurate prediction. The key ideas behind MCNN are to introduce a latent binary network to extract informative regions of interest that contain important signals for prediction and to integrate the latent binary network with CNN to achieve better prediction in various supervised learning problems. Extensive numerical studies demonstrate the competitive performance of the proposed MCNN models. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords: Alzheimer's Disease Neuroimaging Initiative; convolutional neural networks; supervised learning

1. Introduction

In the past decade, convolutional neural network (CNN) models have received much attention due to their competitive performance in various tasks, including object classification (He et al., 2016; Arvidsson et al., 2018; Xie et al., 2020) and semantic segmentation (Badrinarayanan et al., 2017; Chen et al., 2018; Yang et al., 2019; Falk et al., 2019). For instance, in object classification problems, we may be interested in identifying objects in the images that are associated with class labels. Many CNN-based models facilitate learning data-driven, highly representative, layered hierarchical image features from complex datasets; an example is ImageNet (Deng et al., 2009). These models are quite robust

[a]**Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599. Email: yufengliu622@gmail.com.**
[b]**Department of Statistics and Operations Research, Department of Genetics, Department of Biostatistics, Carolina Cancer for Genome Sciences, Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599. Email: yfliu@email.unc.edu.**
[c]**Department of Biostatistics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599. Email: htzhu@email.unc.edu.**

to the large variation of object locations and sizes and tend to aggregate information over the whole image. However, it can be difficult to interpret most high-level image features extracted from CNNs, thus interpretability remains to be a major challenge (Zhang & Zhu, 2018).

To address this challenge, we propose to modify standard CNNs to achieve better model interpretability and prediction in certain supervised learning problems without any additional human supervision. Our proposed approach differs from that of most existing methods in the direction of understanding neural network representations, but can be regarded as a set of neural network models with interpretable/disentangled representations. A comprehensive review of various methods for improving the model's ability to interpret CNNs can be found in Zhang & Zhu (2018).

The aim of this paper is to propose a set of masked CNN (MCNN) models with high model interpretability and better prediction. The key components of MCNN are shown in Figure 1 and summarized as follows:

- Introduce a latent binary network to extract regions of interest (ROIs) for each image that contain informative signals for prediction.
- Simultaneously learn the latent binary network with CNN for achieving better prediction in various supervised learning problems.

Our MCNN can be regarded as a novel extension of the standard two-stage computer-aided diagnostic approach, which consists of segmenting objects of interest (e.g., a tumor) in the first stage and using segmented objects for prediction in the second stage. Specifically, MCNN is a simultaneous segmentation-prediction approach that integrates a semantic segmentation network and CNN into a single neural network model to improve prediction accuracy.

Compared with the existing methods in the literature, three major methodological contributions in this paper are summarized as follows:

- First, MCNN introduces the latent binary network to carry out population semantic segmentation across all samples. It focuses on objects that are highly predictive of the response of interest; whereas standard semantic segmentation networks are developed to identify various ROIs in individual images that represent different objects. Moreover, although semantic segmentation networks are able to deliver pixel-wise annotations, they require extensive human labeling in preparing training samples, which are expensive to acquire (Yu & Koltun, 2015; Long et al., 2015; Chen et al., 2017).
- Second, we propose to learn the latent binary network with CNN to improve its interpretability, which can be widely applicable to CNNs with different architectures. Moreover, MCNN focuses on the informative object learned from the latent binary network by ruling out the irrelevant regions, so it may enhance predictive signals, subsequently leading to better prediction.
- Third, MCNN is applicable to three major types of scenarios that are clustered according to whether accurate annotated mask images are available or not. This is in general enough to cover most real applications.

The rest of this paper is organized as follows. In Section 2, we discuss three different scenarios that MCNN is mainly designed for. In Section 3, we introduce the technical details of the proposed MCNN. We demonstrate the performance of MCNN in two synthetic experiments in Section 4 and two real applications in Section 5. Section 6 concludes the paper with some discussion.

2. Data structure

We consider three different scenarios that commonly appear in real applications.

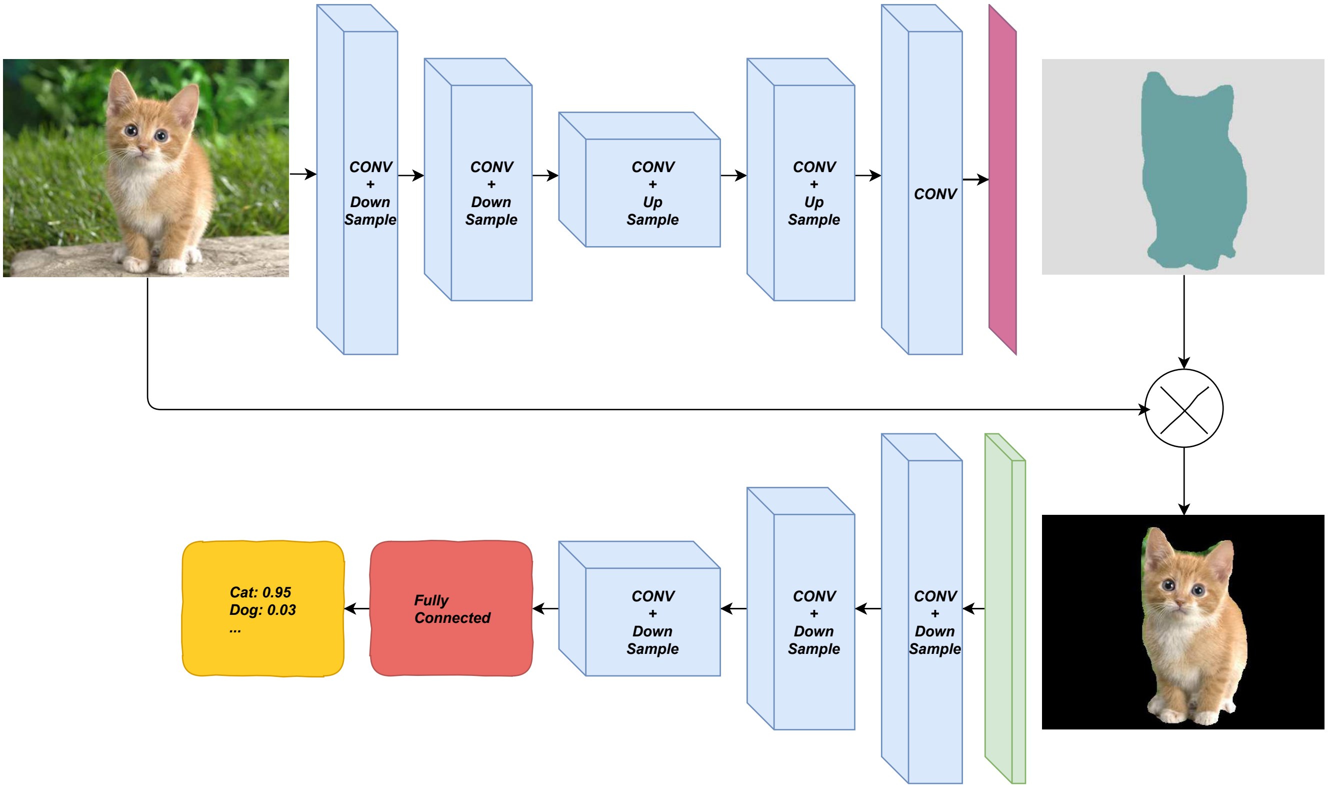

Figure 1. A representative structure of MCNN consisting of the latent binary mapping module with U-net and a classification network based on VGG (Simonyan & Zisserman, 2014). Here $\otimes$ denotes the element-wise product between the masking matrix and the input image.

- Scenario 1 assumes that we observe mask images that accurately capture the true predictive regions of interest.
- Scenario 2 assumes that we observe mask images that roughly capture the true predictive regions. This scenario is in general enough to cover the case without any masked information.
- Scenario 3 assumes that imaging data are mixed either with annotated and unannotated mask images or with high- and low- quality annotations. It can be regarded a mixture of Scenarios 1 and 2.

The proposed MCNN efficiently handles all three scenarios by assigning various weights for different samples. Details about various strategies for handling each scenario are discussed in Section 3.3.

3. Masked convolutional neural network

MCNN consists of two connected modules: a segmentation module that estimates the latent mask maps and a prediction module to perform regression or classification. We denote the network architecture of MCNN as

$$\widehat{y} = \mathcal{F}_1\left(X \otimes \mathcal{F}_0\left(X\right)\right),$$

where X represents the input image; $\mathcal{F}_0$ and $\mathcal{F}_1$ respectively correspond to the segmentation and prediction networks; $\widehat{y}$ denotes the predicted value, which can be a vector of K probabilities for a K-category classification problem or real values in the regression setting; and "$\otimes$" represents the pixel-wise multiplication.

3.1. Segmentation module

The segmentation module enhances the predictive signals by masking off 'background noises'. It estimates a latent binary map (or mask image) $\widehat{M}$ from the input image X such that we have $\widehat{M} = \mathcal{F}_0(X)$. For instance, we set $F_0(\cdot)$ to be the U-net architecture that consists of a symmetric structure of auto-encoders and decoders (Ronneberger et al., 2015). In this case, the module utilizes convolutional and down-sampling layers to aggregate the information over the whole imaging space, gradually expands the feature maps by deconvolutional or up-sampling operations, and eventually constructs a probability map $\widehat{M}$ as the mask. Compared with standard semantic segmentation methods, MCNN does not require pixel-wise annotated images, even though knowing such annotated images may substantially increase the discrimination power. Furthermore, the individual mask images $\widehat{M}$ explicitly localize important ROIs that contribute most to prediction outputs at the pixel level. As shown in numerical examples, the individual mask images $\widehat{M}$ can efficiently handle objects with large variation in terms of both location and size across samples. Thus, the use of $\widehat{M}$ dramatically improves the model's ability to interpret deep neural networks.

3.2. Prediction module

The prediction module $\widehat{y} = \mathcal{F}_1(\widehat{M} \otimes X)$ is directly connected with the output end of the segmentation module. Specifically, it takes the mask images as input and estimates the classification probabilities or numerical response according to the type of learning problems. This module can adopt any CNN architecture. It sequentially processes the mask input images by using convolutional and down-sampling operations and utilizes fully connected layers to estimate the final predicted responses.

3.3. Loss functions

The loss function of MCNN is the weighted sum of a segmentation loss and a prediction loss:

$$l(\widehat{y}, \widehat{M}) = l_p(\widehat{y}) + \lambda l_s(\widehat{M}),$$

where λ balances the two modules. Moreover, in Scenario 1, we have mask images with precise pixel-wise annotations, so we may assign a relatively large λ to emphasize the segmentation module. In Scenario 2 with "imprecise" pixel-wise annotations, we use small values for λ. In the case of mixed samples (Scenario 3), we may apply adaptive weights according to the precision of annotations that is often available in practice.

The segmentation loss measures the similarity between the estimated probability map $\widehat{M}$ and the given mask image M. Popular options include the cross-entropy loss and dice coefficients. The cross-entropy loss measures the similarity between $\widehat{M}$ and M at the pixel level, i.e., $l_s(\hat{M}) = \sum_{1\leq i \leq d_1} \sum_{1 \leq j \leq d_2} M_{i,j} \log \hat{M}_{i,j} + (1 - M_{i,j}) \log \left(1 - \hat{M}_{i,j}\right)$, whereas the dice coefficient loss quantifies the overlap between the estimated map $\widehat{M}$ and the training mask M, and is given by $l_s(\hat{M}) = -\frac{\sum_{i,j} M_{i,j}\hat{M}_{i,j}}{\sum_{i,j} M_{i,j} + \sum_{i,j} \hat{M}_{i,j} - \sum_{i,j} M_{i,j}\hat{M}_{i,j}}$.

The prediction loss varies depending on the type of learning problems. For classification problems, we adopt a cross-entropy loss that measures the Kullback–Leibler divergence between the estimated probabilities and the observed values, i.e., $l_p(\widehat{y}) = -\sum_{k=1}^{K} \{y_k \log \widehat{y}_k + (1 - y_k) \log (1 - \widehat{y}_k)\}$, where $\widehat{y} = [\widehat{y}_1, \ldots, \widehat{y}_K]^T$ represents the vector of predicted probabilities for the K class labels, and y_k is the categorical indicator, i.e., $y_k = 1$ if the object belongs to the k-th category. For regression problems, we may apply the squared loss, i.e., $l_p(\widehat{y}) = (\widehat{y} - y)^2$.

3.4. Implementation

We adopt a U-net (Ronneberger et al., 2015) structure as our segmentation module. We use a convolutional layer with kernel size $(3, 3)$ or $(3, 3, 3)$ for 2D or 3D input images, respectively, and apply zeros as padding to maintain fixed image sizes during convolutions. We add batch normalization (Ioffe & Szegedy, 2015) layers after each convolutional layer and activate the feature maps using rectifier liner units (Nair & Hinton, 2010). In the down-sampling phase of the network, we apply the maximum pooling layers with kernel size $(2, 2)$ or $(2, 2, 2)$, and use repetition up-sampling with the same kernel size to increase the resolution of feature maps in the up-sampling phase. The segmentation loss is the pixel wise cross-entropy function.

For the prediction module, we use a VGG structure (Simonyan & Zisserman, 2014) with batch normalized convolutional layers for 2D images, and a ResNet structure (He et al., 2016) for 3D images. For 2D images, we use the $\lambda = 10^{-4}$ as the weight parameter of the segmentation module and $\lambda = 10^{-6}$ for 3D images when rough annotations are provided.

We conduct the numerical experiments with training, validation, and test datasets, and use the stochastic gradient descent algorithm to train the networks (see (Ruder, 2016) for details about the algorithm). In particular, we use training sets to estimate model parameters and evaluate the model on the validation set at the end of each epoch. We initialize the learning rate of the algorithm with 10^{-4} and gradually decrease it if the validation loss does not decrease for ten consecutive epochs. The model with minimum validation loss is outputed as the final model and is used to evaluate the predictive accuracy on the test set.

In the numerical studies, we compare our model with the corresponding prediction network without the segmentation module, i.e., the model with the same structure as the prediction module in MCNN. We denote such a model as CNN to distinguish it from MCNN. For Scenarios 1-3, the MCNN models are denoted as MCNN 1-3 respectively, i.e., MCNN 1 for the case with precise annotations, MCNN 2 for imprecise annotations and MCNN 3 for mixed samples.

4. Synthetic simulation experiments

In this section, we conduct numerical studies with synthetic images, in which the segmentation ground truth is known. We generate the training mask according to the three scenarios discussed in Section 2, and apply the strategies in Section 3 to assign the loss weight.

4.1. Synthetic image regression

In this experiment, we simulate a set of symbolic images, each containing 3 ROIs: a circle, a square, and a triangular region. The ROIs vary randomly by size and by location within a 32-by-32-pixel grid. The responses are generated according to the radius of the circles and the area of the squares. The triangular regions are not associated with the responses. We add Gaussian noise with standard deviation 1 to each response and impose background noise to the covariate images according to a Gaussian random field. We generate $40,000$ training, $10,000$ validation, and $10,000$ test samples. The precise training masks in Scenario 1 are the images with pixel-level annotations of the corresponding ROIs. The rough mask images in Scenario 2 cover the predictive signals with larger irregular regions. For scenario 3, only 20% of the training samples have pixel-wise annotations, and the rest do not have any annotation information.

Some results are illustrated in Figure 2. The estimated masks clearly capture the predictive ROIs while ruling out the irrelevant triangular region when precise annotations are provided. With rough training masks, the predictive ROIs are still identifiable by the model and the background noise are reduced. In each of the three scenarios, the non-predictive triangular regions are effectively ruled out.

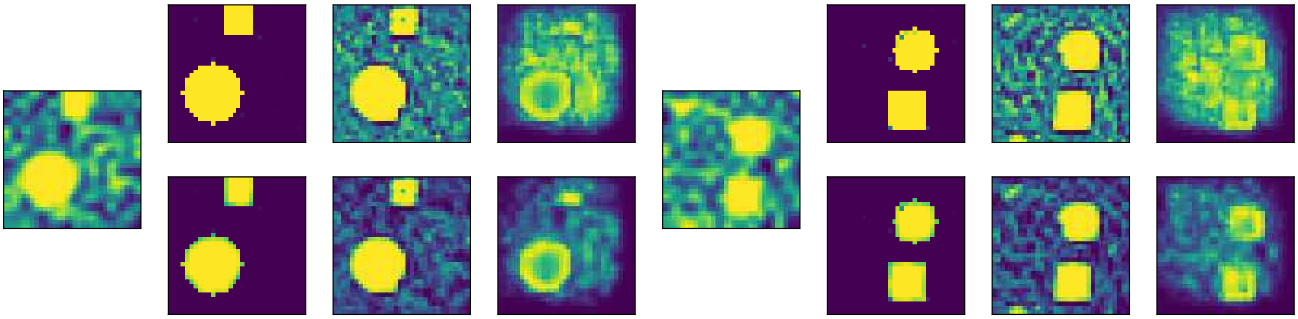

Figure 2. Estimation results for the synthetic image regression. In each of the two panels, the image on the left is the original input image. The three images on the top are the estimated masks for Scenario 1-3 respectively. The images on the bottom are the corresponding mask images.

4.2. Noisy MNIST

The MNIST (LeCun et al., 1998) dataset was constructed from a number of scanned documents collected by the National Institute of Standards and Technology. Each image is of size 28 by 28 pixels. There are $50,000$ training samples, $10,000$ validation samples, and $10,000$ test samples in the dataset. The original MNIST images have clean backgrounds and are relatively easy to classify with high accuracy, e.g., over 99.5% in Wan et al. (2013). In order to evaluate the improvement of the proposed network structure, we conducted our experiment with noisy MNIST images that we created by adding random noise to the original MNIST images and randomly resizing the digits and shifting them within a 32-by-32-pixel grid. The precise training masks cover the digit regions in each image. The rough masks contain broader regions around the digits. For mixed samples, we assign 20% of the training samples with pixel-wise annotations, and the rest with no annotation information.

Figure 3 illustrates some of the estimation results, from which we can see that the estimated mask images can effectively block out background noise and accurately extract signals of digits from the noisy images. With the precise annotations, the extracted ROIs are more accurate, even when such annotations are available only for a small portion of samples.

5. Real applications

We apply MCNN models in real image prediction problems, in which the ground truth of the mask images are unknown. We apply our prior knowledge about the data and create binary images that roughly cover the predictive regions as training mask images. Those masks are unique across different samples and are refined accordingly in the estimated results. Therefore, these experiments can be categorized to Scenario 2 in Section 2.

5.1. Street view house number (SVHN)

In this experiment, we classify the street view house number (SVHN) images (Netzer et al., 2011). This dataset consists of color images of house numbers collected by Google Street View. Each image is of size 32×32 pixels and may contain multiple digits. Our target is to classify the digit in the center of the image. The other digits must be ignored. There are $73,257$ images in the training set and $26,032$ images in the test set. We further split the training images into $50,000$ training samples and $23,257$ validation samples for model selection. We applied the training mask

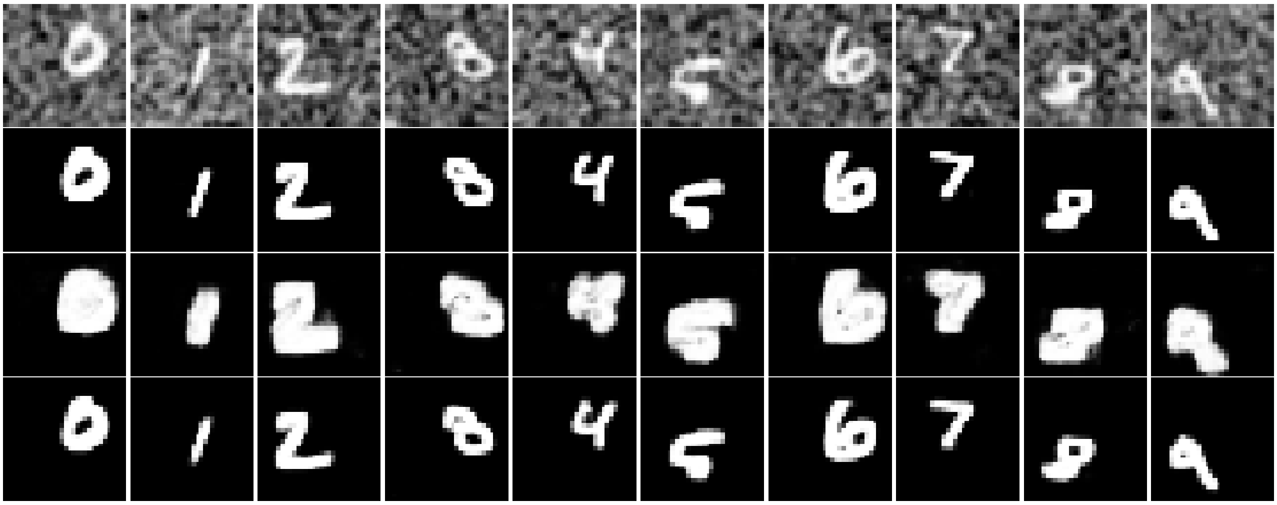

Figure 3. The estimation results for the noisy MNIST experiment. The top row is the plot of the input images with digits 0 to 9 from left to right. The following rows are the corresponding estimated masks from Scenario 1-3 respectively.

images that cover the middle part and ruled out the side regions of the input images. With such mask images, the model tends to focus on the center of the images so that any interference from the digits on the side can be reduced.

We illustrate the estimation results in Figure 4 and the training history in Figure 5. The estimated masks are able to highlight the target digits in the center while masking the digits on the sides. The test accuracy is improved from 90.73% to 95.13% compared to that achieved by the CNN model. Some of the incorrect classification of images is potentially due to incorrectly annotated labels (e.g., the first row in the right group of images) or incomplete digit regions (see the second row in the right group). The training history also indicates a stable validation error and improved classification accuracy.

5.2. ADNI MRI classification

In this experiment, we aim to classify the structural magnetic resonance imaging (MRI) data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) study (see Figure 6). The ADNI study was launched in 2003 as a large-scale, long-term project to collect MRI and positron emission tomography images, cerebrospinal fluid, and blood biomarkers, among other data. The goal of the ADNI study is to track the progression of Alzheimer's disease using these biomarkers and assess the brain's structural and functional changes over different disease states. More information about this study can be found at the ADNI website (http://www.adni-info.org/).

We utilize the RAVENS-maps of the T1-weighted MRI images from different phases of the study, including ADNI1, ADNI2 and ADNI GO. The total number of participants in this study is 749, including participants with Alzheimer's disease, mild cognitive impairment, and healthy status. For each participant, multiple images are collected at different time points, and their disease status may vary as well. We use the disease status at the time of image acquisition as its corresponding class label. After dropping the images with no diagnostic results or low quality, we collect a total of $3,021$ images in our study. We generate the RAVENS-maps by following the pipeline in Liu et al. (2018) and further down-sample the maps to the resolution of $64 \times 64 \times 64$ for the consideration of computational load. We randomly

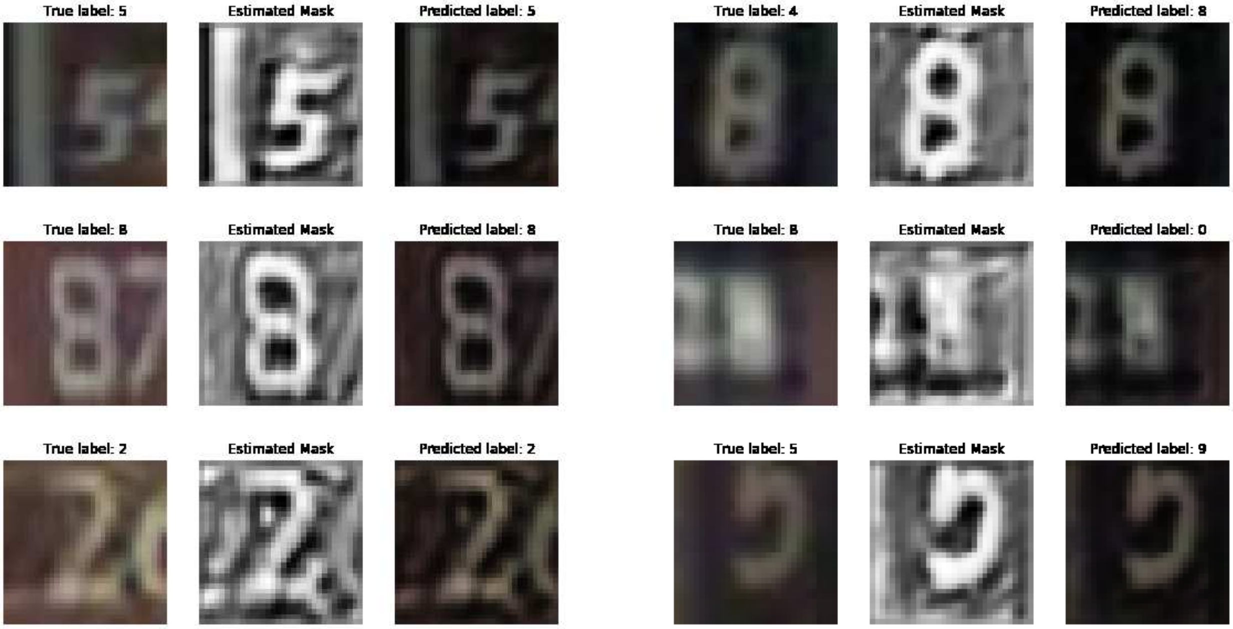

Figure 4. The estimation results for the SVHN experiment. The left group of images shows three correctly classified images, and the right group corresponds to three misclassified images. Each of the six panels consists of three images: the original noisy image, the estimated mask and the mask image.

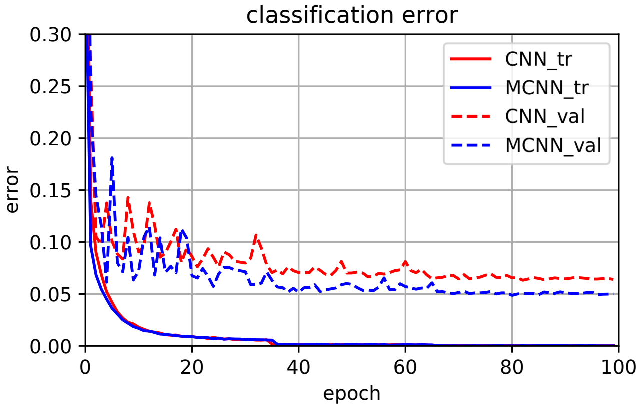

Figure 5. Training history of the CNN and MCNN models in the SVHN experiment. The red and blue lines represent the loss function value of the CNN and MCNN models, respectively. The solid and dashed lines correspond to the training and validation losses, respectively.

split the samples into training (80%), validation (10%), and test (10%) sets in the modeling procedure, and apply a mask that covers the whole brain region as the training image for each sample.

This experiment involves 3D images, and all the networks are built with 3D operations, including 3D convolution, up-sampling and pooling. In consideration of the model size, we use the ResNet structure (He et al., 2016) for the

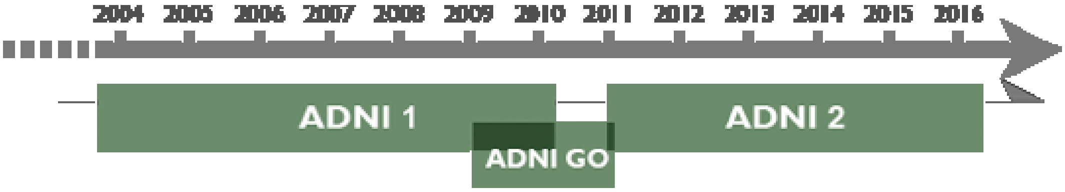

Figure 6. Timeline of the ADNI1, ADNI2 and ADNI GO studies. Image source: $http://adni.loni.usc.edu/about/$.

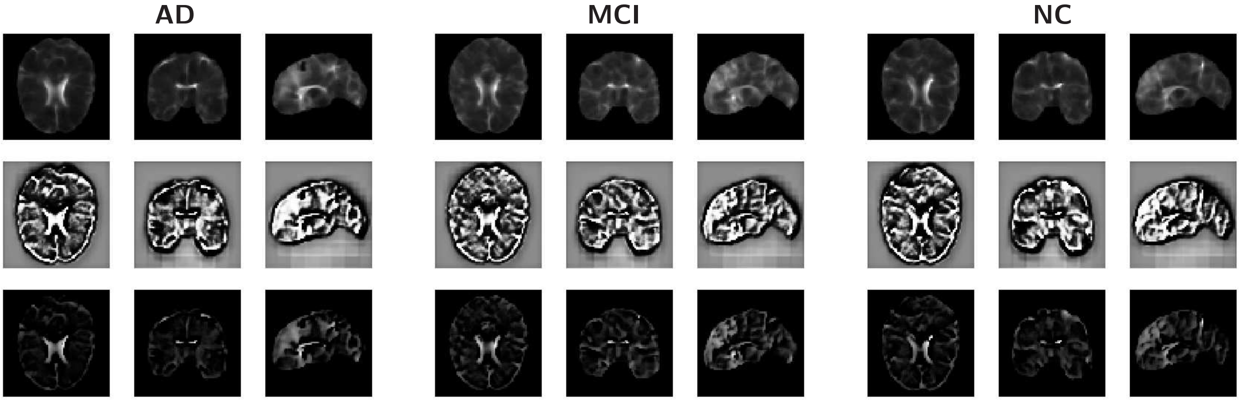

Figure 7. Estimation results for the ADNI experiment. The three groups of brain images respectively show typical samples of Alzheimer's disease (AD), mild cognitive impairment (MCI) and healthy status (NC). From top to bottom, each column shows three images: the original image, the estimated mask, and the mask image.

prediction phase.

Figure 7 illustrates some of the estimation results. We can see that the estimated masks tend to select most of the brain regions, while focusing on the frontal cortex, temporal gyrus, hippocampus, and fornix regions. These parts of the brain have been well studied in the literature and have been shown to relate to planning, logical thinking, and memory (Bordi et al., 2016; Lozano et al., 2016; Association, 2019). With such masks, the classification accuracy is improved from 90.04% to 92.74% compared to that of the CNN model. From the plot of training losses in Figure 8, we can see that the loss of MCNN decreases more slowly at the beginning, but decreases more quickly after 60 epochs, and the validation results become more stable as well. This is mainly due to the larger model size compared to that of the CNN model. Once a stable estimation from the segmentation module is achieved, the classification results become better and more stable.

We summarize the prediction results of all the experiments in Table 1. Note that with the segmentation module added, the MCNN models tend to deliver better results in both regression and classification. Moreover, the more precise annotations, the better prediction we can have.

6. Discussion

In this paper, we propose an MCNN model that can carry out simultaneous segmentation-prediction procedures. More importantly, the proposed model can generate mask images that are able to select the predictive ROIs in the input images and mask off the background noise. This can potentially enhance the target signals and improve the predictive

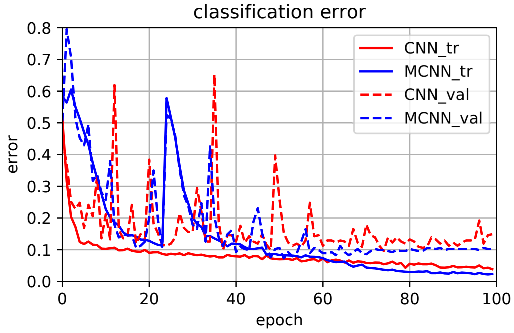

Figure 8. Training history of the CNN and MCNN models in the ADNI experiment. The red and blue lines represent the loss function values of the CNN and MCNN models, respectively. The solid and dashed lines correspond to the training and validation losses, respectively.

Table 1. Summary of results in the numerical experiments. The mean squared prediction errors are reported for the regression problem and the misclassification errors are reported for the classification problems.

Method	SIM	MNIST	SVHN	ADNI
CNN	2.27	5.30%	9.27%	9.60%
MCNN1	**1.31**	**2.02%**	N/A	N/A
MCNN2	1.60	2.33%	**4.87%**	**7.26%**
MCNN3	1.51	2.28%	N/A	N/A

accuracy.

The segmentation module in the MCNN model functions as the pre-whitening process for the input images. This is beneficial when the backgrounds are actual noise and not informative for the prediction. In some cases, if the background signals and target objects were highly correlated, then MCNN might not significantly improve the prediction.

We have focused on cases with non-informative backgrounds in the numerical experiments. The main purpose of these studies is to demonstrate the improved prediction achieved by MCNN due to the segmentation module. Thus, instead of comparing our proposed model's results with benchmark results, we have mainly compared the MCNN models with their corresponding CNN models with the same structure as our prediction module. We do not have a specific requirement for the network structure. Any segmentation and prediction networks can be used to construct MCNN. However, including the segmentation module increases the size of the model. This can be a potential issue, but it may be solved by parameter sharing, i.e., using the same structure and parameters in the down-sampling phase of both the segmentation and prediction modules. The competitive performance of MCNN both in terms of interpretability and accuracy suggests that this is a promising area for future research.

Data Availability Statement

The data that support the findings of this study are openly available on the ADNI website: http://adni.loni.usc.edu/.

References

Arvidsson, I, Overgaard, NC, Marginean, FE, Krzyzanowska, A, Bjartell, A, Åström, K & Heyden, A (2018), 'Generalization of prostate cancer classification for multiple sites using deep learning,' in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, IEEE, pp. 191–194.

Association, A (2019), '2019 alzheimer's disease facts and figures,' *Alzheimer's & Dementia*, **15**(3), pp. 321–387.

Badrinarayanan, V, Kendall, A & Cipolla, R (2017), 'Segnet: A deep convolutional encoder-decoder architecture for image segmentation,' *IEEE transactions on pattern analysis and machine intelligence*, **39**(12), pp. 2481–2495.

Bordi, M, Berg, MJ, Mohan, PS, Peterhoff, CM, Alldred, MJ, Che, S, Ginsberg, SD & Nixon, RA (2016), 'Autophagy flux in ca1 neurons of alzheimer hippocampus: Increased induction overburdens failing lysosomes to propel neuritic dystrophy,' *Autophagy*, **12**(12), pp. 2467–2483.

Chen, LC, Papandreou, G, Kokkinos, I, Murphy, K & Yuille, AL (2018), 'Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,' *IEEE transactions on pattern analysis and machine intelligence*, **40**(4), pp. 834–848.

Chen, LC, Papandreou, G, Schroff, F & Adam, H (2017), 'Rethinking atrous convolution for semantic image segmentation,' *arXiv preprint arXiv:1706.05587*.

Deng, J, Dong, W, Socher, R, Li, LJ, Li, K & Fei-Fei, L (2009), 'Imagenet: A large-scale hierarchical image database,' in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, pp. 248–255.

Falk, T, Mai, D, Bensch, R, Çiçek, Ö, Abdulkadir, A, Marrakchi, Y, Böhm, A, Deubner, J, Jäckel, Z, Seiwald, K et al. (2019), 'U-net: deep learning for cell counting, detection, and morphometry,' *Nature methods*, **16**(1), pp. 67–70.

He, K, Zhang, X, Ren, S & Sun, J (2016), 'Deep residual learning for image recognition,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Ioffe, S & Szegedy, C (2015), 'Batch normalization: Accelerating deep network training by reducing internal covariate shift,' *arXiv preprint arXiv:1502.03167*.

LeCun, Y, Bottou, L, Bengio, Y & Haffner, P (1998), 'Gradient-based learning applied to document recognition,' *Proceedings of the IEEE*, **86**(11), pp. 2278–2324.

Liu, LYF, Liu, Y, Zhu, H, Initiative, ADN et al. (2018), 'Smac: Spatial multi-category angle-based classifier for high-dimensional neuroimaging data,' *NeuroImage*.

Long, J, Shelhamer, E & Darrell, T (2015), 'Fully convolutional networks for semantic segmentation,' in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440.

Lozano, AM, Fosdick, L, Chakravarty, MM, Leoutsakos, JM, Munro, C, Oh, E, Drake, KE, Lyman, CH, Rosenberg, PB, Anderson, WS et al. (2016), 'A phase ii study of fornix deep brain stimulation in mild alzheimer's disease,' *Journal of Alzheimer's Disease*, **54**(2), pp. 777–787.

Nair, V & Hinton, GE (2010), 'Rectified linear units improve restricted boltzmann machines,' in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Netzer, Y, Wang, T, Coates, A, Bissacco, A, Wu, B & Ng, AY (2011), 'Reading digits in natural images with unsupervised feature learning,' in *NIPS workshop on deep learning and unsupervised feature learning*, 2, p. 5.

Ronneberger, O, Fischer, P & Brox, T (2015), 'U-net: Convolutional networks for biomedical image segmentation,' in *International Conference on Medical image computing and computer-assisted intervention*, Springer, pp. 234–241.

Ruder, S (2016), 'An overview of gradient descent optimization algorithms,' *arXiv preprint arXiv:1609.04747*.

Simonyan, K & Zisserman, A (2014), 'Very deep convolutional networks for large-scale image recognition,' *arXiv preprint arXiv:1409.1556*.

Wan, L, Zeiler, M, Zhang, S, Le Cun, Y & Fergus, R (2013), 'Regularization of neural networks using dropconnect,' in *International Conference on Machine Learning*, pp. 1058–1066.

Xie, H, Wang, N, He, M, Zhang, L, Cai, H, Xian, J, Lin, M, Zheng, J & Yang, Y (2020), 'Using deep learning algorithms to classify fetal brain ultrasound images as normal or abnormal,' *Ultrasound in Obstetrics & Gynecology*.

Yang, H, Shan, C, Kolen, AF & de With Peter, H (2019), 'Improving catheter segmentation & localization in 3d cardiac ultrasound using direction-fused fcn,' in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, IEEE, pp. 1122–1126.

Yu, F & Koltun, V (2015), 'Multi-scale context aggregation by dilated convolutions,' *arXiv preprint arXiv:1511.07122*.

Zhang, Q & Zhu, S (2018), 'Visual interpretability for deep learning: a survey,' *Frontiers of Information Technology & Electronic Engineering*, **19**, pp. 27–39.