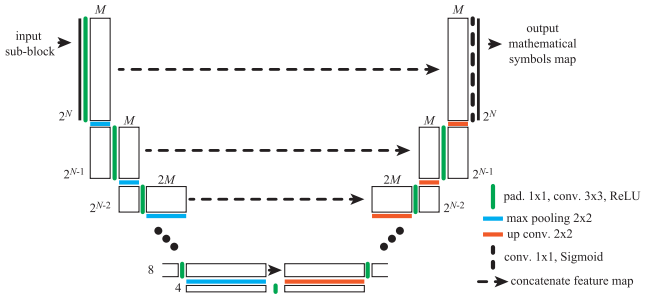


FIGURE 4. The U-Net architecture in the proposed method.

symbols are white. To prevent the elimination of thin components in the document image, white regions are dilated by 1 pixel in each direction using the mathematical morphology operation.

Overlapped square sub-blocks are defined to cover entire image regions and extracted to be the inputs to the image conversion stage. The edges of the input document image are wrapped by the opposite side of the image when the sub-block is over the image edges. The sub-block operation is used initially to deal with the memory size limits [7]. Also, the sub-block operation plays a role as data augmentation without image deformation operations. Generally, ideal estimation of variations of data is crucial for designing data augmentation protocols. The proposed method assumes that a flat-bed scanner captures the input images so that the input images do not contain significant image deformation. Consequently, data augmentation with such nonlinear image deformation is not required.

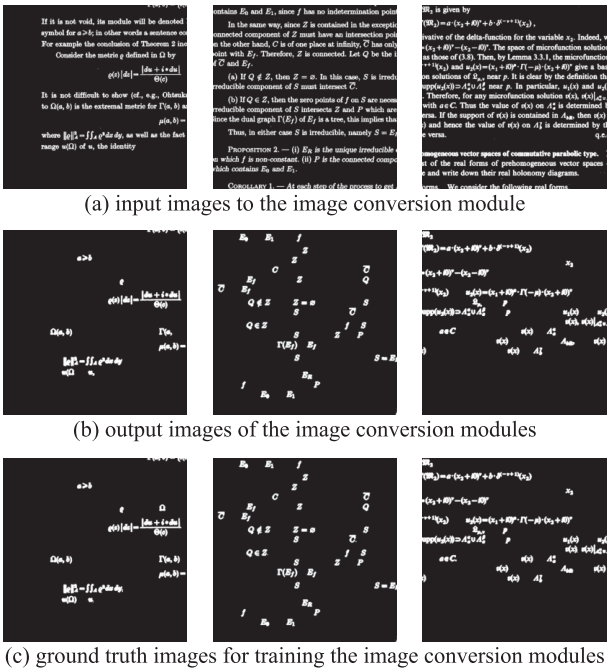
The width and height of the sub-blocks are parameters that affect the performance of the proposed method. The actual size of the sub-block images in our implementation was determined by the results of a preliminary experiment discussed in IV-C

B. ME DETECTION USING U-NET

ME detection in the proposed method can be considered as an image conversion task. Figure 5 shows examples of input, output and ground truth images of the ME detection process. As shown in the figure, the ME detection process is required to eliminate regions from MEs and extract the CCs that construct MEs.

We use the U-Net architecture proposed by Ronneberger *et al.* [7], motivated by the promising achievement of its semantic segmentation of biomedical images. U-Net is an FCN architecture that was proposed for the segmentation of biomedical images. By introducing skip connections between corresponding layers in the encoder and decoder, it successfully preserves the high-frequency components in the converted output images.

Figure 4 shows the actual U-Net configuration in the proposed method. The network mainly consists of two stages, i.e encoding and decoding stages. In encoding stage, the typical CNN architecture is employed. The encoding stage consists of multiple applications of a 3×3 convolution with a



(a) input images to the image conversion module

(b) output images of the image conversion modules

(c) ground truth images for training the image conversion modules

FIGURE 5. Examples of input, output and ground truth images for image conversion using U-Net.

1×1 padding followed by a rectified linear unit (ReLU) activate function and a 2×2 max-pooling operation for down sampling. The number of feature maps is doubled at each two downsampling steps. The decoding stages consists of an upsampling of the feature map followed by a 2x2 up-convolution. While the concatenation of feature maps in the original U-Net requires the cropping operation because there is loss of border pixels in every convolution, the proposed method does not employ cropping because the overlapped sub-blocks can recover the loss to each other. The final layer employs a 1x1 convolution to map each M-component feature vector to a binary output image.

As shown by Figure 4, the proposed method assumes that the size of an input sub-block is determined by $2^N \times 2^N$. The number of layers in the encoder and decoder stages is corresponding to the sub-block size. The base number of feature maps $M = 64$ is determined from the original U-Net impremention.

We implemented and trained U-Net to convert the input sub-block image to an image that contained only the CCs that constructed MEs. To achieve this conversion, we created ground truth images using an annotated dataset. In the dataset, we determined whether each character was a mathematical symbol or ordinary character. We eliminated the CCs annotated as ordinary characters to create the ground truth images. The ground truth in the training dataset was a set of sub-block images that were extracted at the corresponding position on the input image. We used the Dice loss determined by the following as the objective loss function to be minimized because the task of U-Net is binary-to-binary image conversion:

$$L(X, Y) = 1 - D(X, Y) = 1 - \frac{2|X \cap Y|}{|X| + |Y|}, \qquad (1)$$