

# A Digital Protein Microarray for COVID-19 Cytokine Storm Monitoring

Yujing Song<sup>1</sup>, Yuxuan Ye<sup>1</sup>, Shiuan-Haur Su<sup>1</sup>, Andrew Stephens<sup>1</sup>, Tao Cai<sup>1</sup>, Meng-Ting Chung<sup>1</sup>, Meilan Han<sup>2</sup>, Michael W. Newstead<sup>2</sup>, Lenar Yessayan<sup>3</sup>, David Frame<sup>4</sup>, David Humes<sup>3</sup>, Benjamin H. Singer<sup>2,6\*</sup>, and Katsuo Kurabayashi<sup>1,5,6\*</sup>

<sup>1</sup>Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, 48109, United States.

<sup>2</sup>Department of Internal Medicine, Division of Pulmonary and Critical Care Medicine, University of Michigan, Ann Arbor, MI, 48109, United States

<sup>3</sup>Department of Internal Medicine, Division of Nephrology, University of Michigan, Ann Arbor, MI, 48109, United States

<sup>4</sup>Department of Clinical Pharmacy, College of Pharmacy, University of Michigan, Ann Arbor, MI, 48109, United States.

<sup>5</sup>Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109, United States

<sup>6</sup>Michigan Center for Integrative Research in Critical Care, University of Michigan, Ann Arbor, MI, 48109, United States

\*Corresponding author:

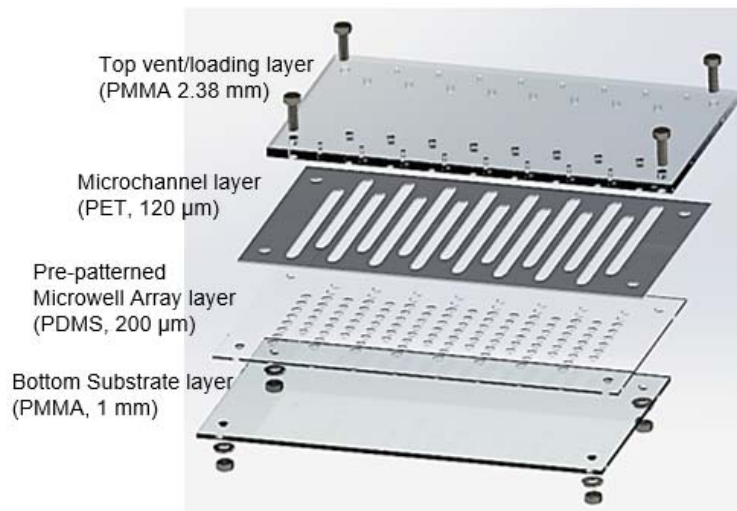
Dr. Katsuo Kurabayashi

E-mail address: [katsuo@umich.edu](mailto:katsuo@umich.edu)

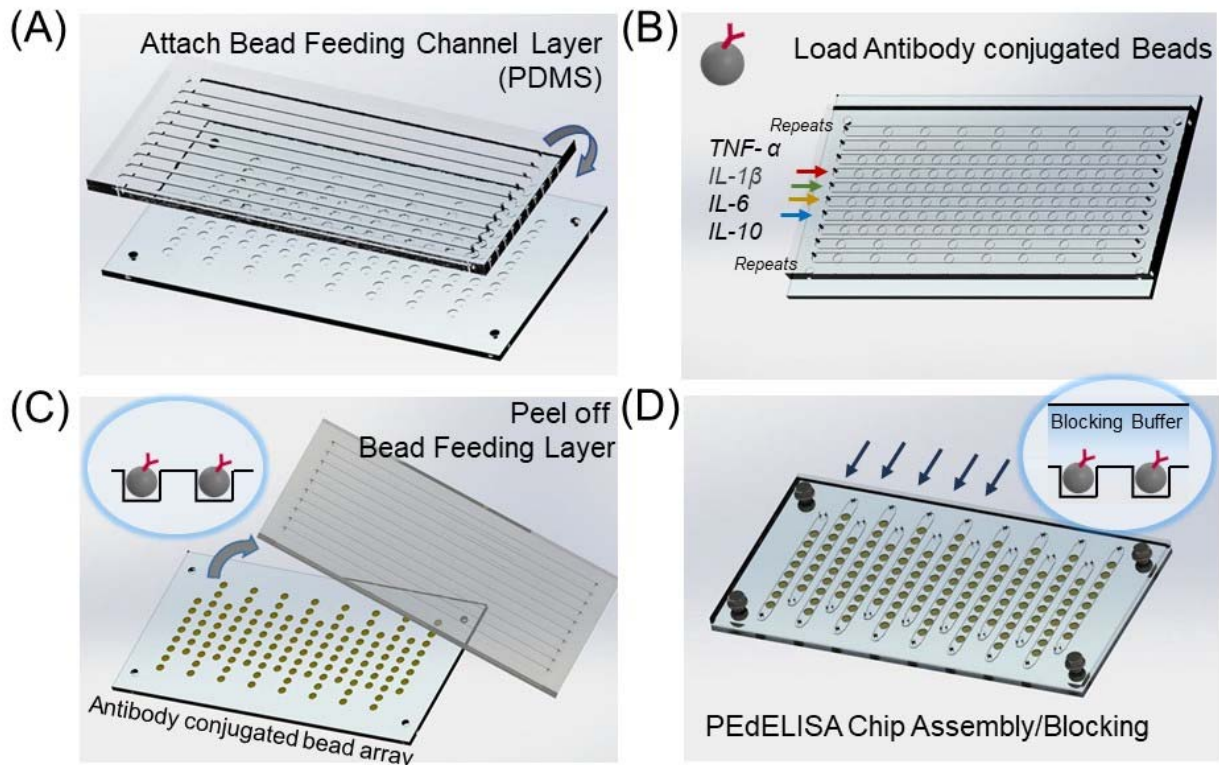
Dr. Benjamin H. Singer

E-mail address: [singerb@med.umich.edu](mailto:singerb@med.umich.edu)

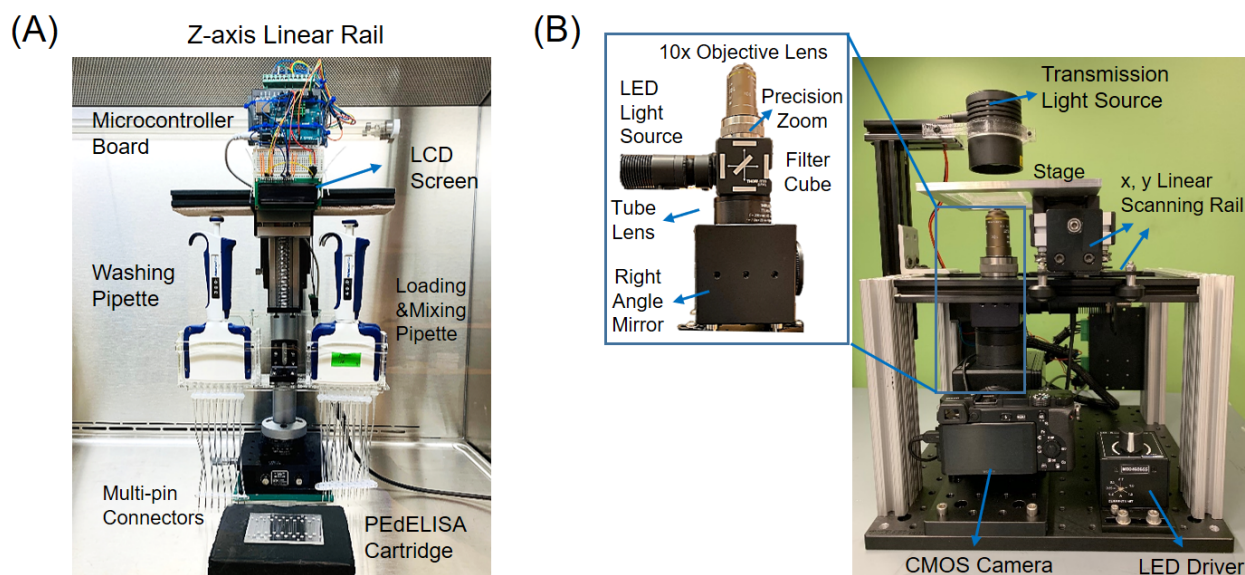
## Supplementary Information



**Figure S1.** Fabrication of disposable PEDELISA microfluidic chip. The chip contains two polymethyl methacrylate (PMMA) layers (top venting and bottom substrate layers), a thin polydimethylsiloxane (PDMS) layer (200  $\mu\text{m}$ ), which contains biosensor patterns, each consisting of fL-sized microwell arrays for digital assay, and a polyethylene terephthalate (PET) thin film (120  $\mu\text{m}$ ) with microfluidic channels fabricated by laser cutting.



**Figure S2.** PEDELISA chip preparation and assembly processes . (A) Attach a PDMS bead feeding channel layer to cover PDMS microwell arrays constructed on the bottom PMMA substrate. (B) Load a buffer solution suspending antibody-conjugated beads into the bead feeding channels. (C) Allow beads to be settled in microwells for 5 min and then gently peel off the bead feeding channel layer. (D) Place the PET sample detection channel layer onto the remaining PMMA substrate and subsequently place the PMMA sample vent/loading layer on top of the PET layer. Assemble the chip using bolt screws (M2). Finally, prime each sample detection channel with blocking buffer (SuperBlock™ PBS buffer, 0.05% Tween20) to passivate the channel surface.



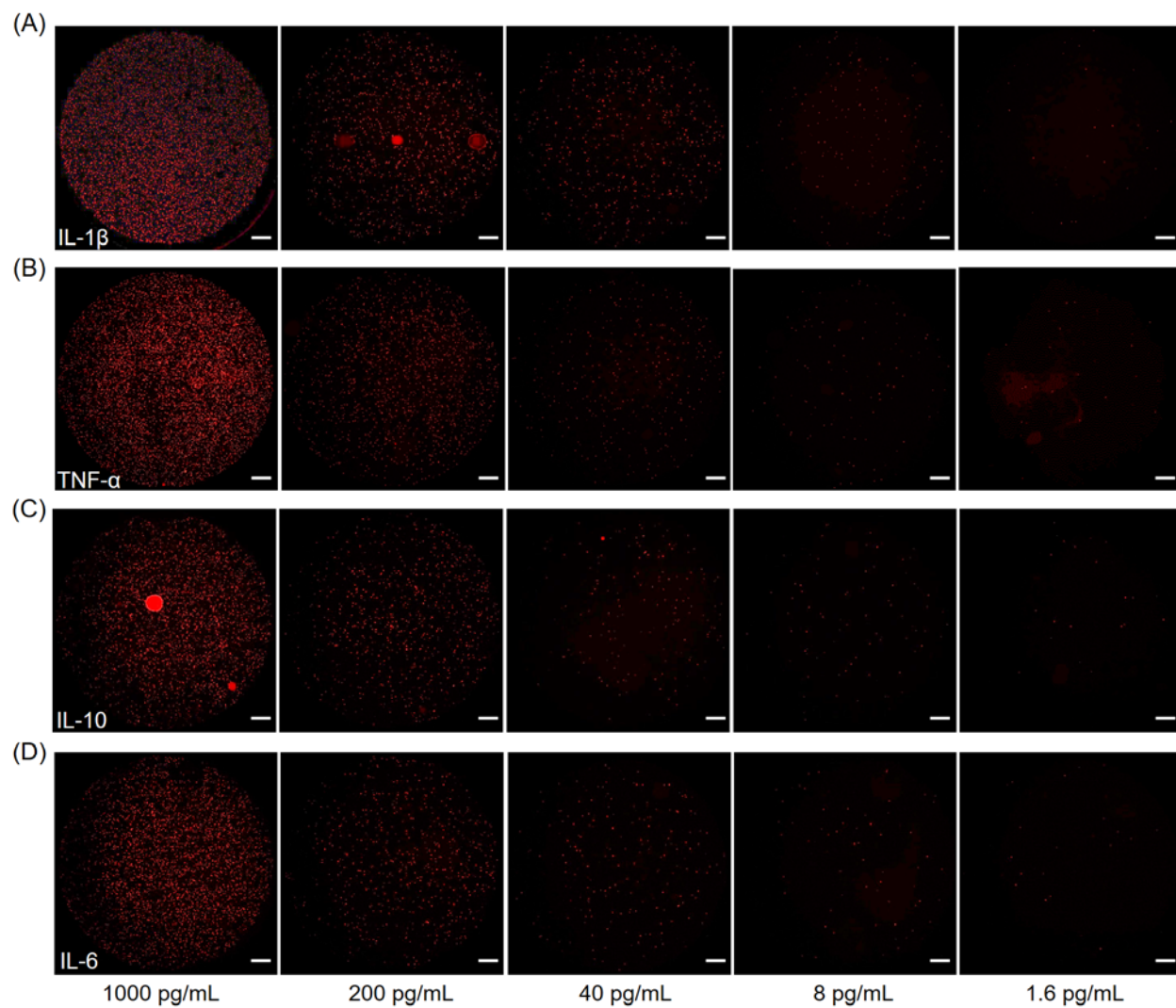
**Figure S3.** PEDELISA assay platform comprising two modules: (A) programmed fluidic handling module with Arduino controlled linear rail and multi-channel pipettes. (B) customized low-cost 2-axis fluorescence scanning module with a consumer-grade CMOS camera.

### **Design of programmed fluidic handling and low-cost optical scanning modules.**

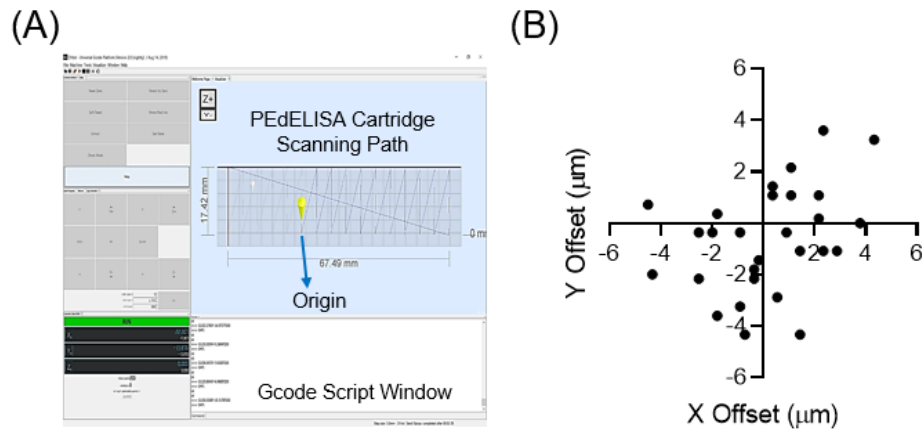
A programmed fluid handling module was designed to allow semi-automated parallel fluid handling (up to 16 samples) and to improve the fidelity and sensitivity of the assay by active on-chip mixing. The module incorporates a linear rail system (one NEMA 17 stepper motor, TB6600 motor shield, Arduino Uno, pipette holders) to operate up to four multi-channel pipettes for assay reagents loading, mixing and washing (Fig. S3A). A multi-pin connector was designed to provide good sealing for the countersink holes on the cartridge. This helps avoid leaking and bubble generation during sample/reagent solution loading. A liquid crystal display (LCD) screen was attached to inform the user of step-by-step instructions with time countdown for each assay step.

An optical scanning module was designed to be cost-effective and compact for high-throughput, high-quality digital image readout. The module comprises a 2D image scanner and an optical unit. The scanner employs a two-axis linear rail system with two 0.9° NEMA 17 stepper motors, which are powered by DM542T motor shields and controlled by an Arduino MEGA 2560 board, a lead screw/rail system, and an imaging stage made of aluminum (Fig. S3B). The optical unit (Fig. S3B inset) was assembled beneath the scanning unit with a 10X Nikon objective lens

mounted on a high-precision non-rotating zoom housing, a 30mm filter cube including a dichroic mirror (565 nm long-path), and an excitation (545/25 nm) and emission filter (605/70 nm). The excitation side of the filter cube was connected to an aspherical lens (F=16 mm) that collimates an LED fluorescence light source (560nm) used for fluorescent excitation of beads. The bottom of the cage cube was connected to a sliding emission filter mount for switching between the brightfield and fluorescent imaging modes. The filter mount was connected to a 200mm tube lens (Nikon) and the light path was reflected towards the side using a 50 mm right-angle prism mirror mounted in a 60mm cage cube. A CMOS sensor (SONY a6100 CMOS camera) was mounted on the side of the 60mm cage cube and remotely controlled by a laptop to perform image scanning as well as data processing. Lastly, a collimated halogen lamp for bright field imaging was mounted above the chip to provide bright field illumination. To achieve a large field and depth of view in imaging as well as increase the compactness of the module, we customized the optical distance between the tube lens and the CMOS plane so that it lowered the magnification ratio of the whole optical system to 9X. To avoid high cost and complexity for the module, we did not use Köhler illumination for brightfield imaging. Instead, all images were post-processed with a simple flatfield correction algorithm using MATLAB. As a result, the cost of the entire optical unit is less than \$4000.



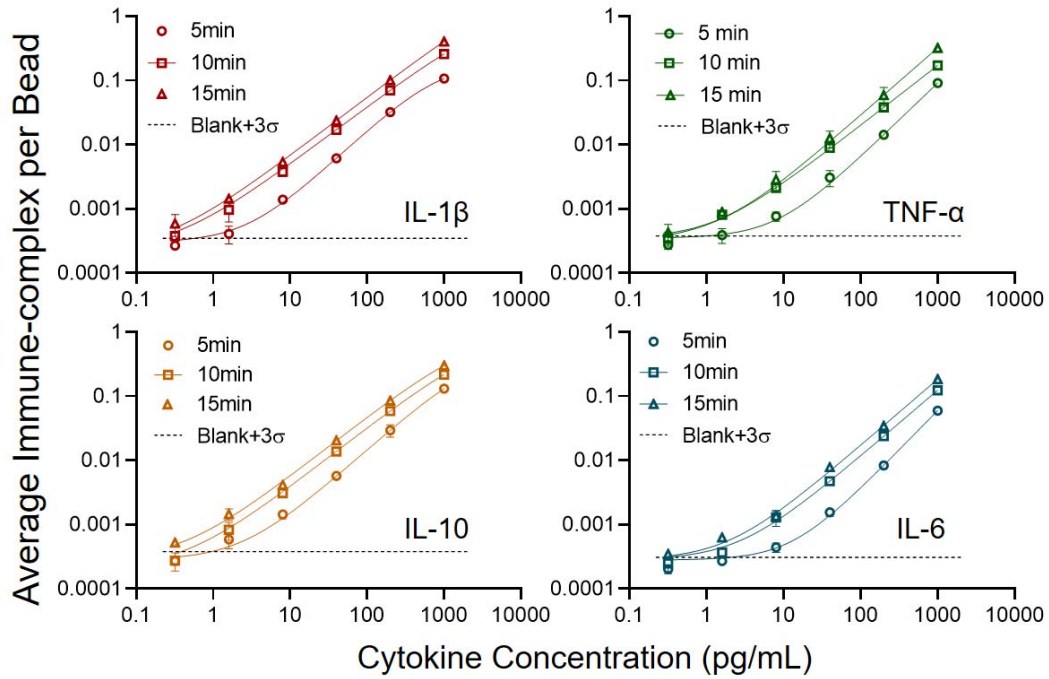
**Figure S4.** Representative snapshot images of enzyme active “On” microwells on a biosensor pattern (66,724 wells/biosensor) for various analyte concentrations of (A) IL-1 $\beta$  (B) TNF- $\alpha$  (C) IL-10 (D) IL-6. For clear visualization, images of 3600 $\times$ 3600 pixels were cropped from original raw images of 6000 $\times$ 4000 pixels with 80% brightness enhancement and 40% contrast enhancement. All of the scale bars are 200  $\mu$ m.



**Figure S5.** (A) User-interface of the Universal Gcode Sender for automated PEDELISA assay image scanning (B) Characterization of bi-directional motion control accuracy by repeatedly scanning and imaging the designated microarray on the chip. Bi-directional repeatability with a positional error of less than 5  $\mu\text{m}$  was achieved.

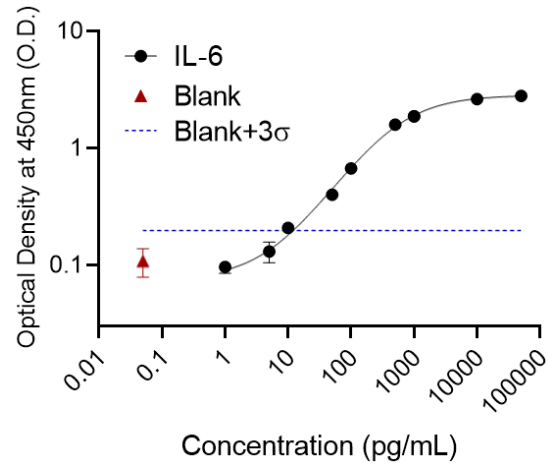
### Control and characterization of 2D optical scanning module

To control the optical scanning module, we use the Universal Gcode Sender (UGS) that transmits a MATLAB-generated Gcode to the Arduino for imaging scanning (Fig. S5A). UGS provides a platform with manual and automated precision control of the stage and visualization of the scanning paths, making the whole module practical for this application. By designating each motor of the system through the Arduino, specific commands can be sent to each motor to move the stage to the desired location. A single command for moving a motor in Gcode includes the origin and units of movement, the shape of travel, the axis and distance of movement, and the motor speed. For the image scanning, an algorithm has been developed to account for any rotational offset in the cartridge placement to ensure that the microscope optics can capture an image of each microwell in the center of the view. We used a MATLAB script that generates a Gcode file containing line-by-line commands between two locations on the cartridge with a 5-sec delay for image focusing and capture. Our preliminary test showed that simultaneous x and y travel increases the errors of the motors. As such, we executed x and y-axis movements individually to improve the accuracy of the scanning. We characterized the x-y motion control accuracy by repeating the scanning and imaging of the on-chip microarray structures. We then used post-image overlay to calculate the x, y offsets which are plotted in Fig. S5B. We were able to achieve repeatable bi-directional scanning with a positional error of less than 5  $\mu\text{m}$  and a minimum incremental movement of 0.31  $\mu\text{m}$ . This provided sufficient accuracy for the PEDELISA assay.



**Figure S6.** Pre-equilibrium assay reaction optimization. PEDELISA standard curves for IL-1 $\beta$ , TNF- $\alpha$ , IL-10, and IL-6, with the incubation time of analyte/detection antibody reaction varying from 5 min, 10 min, 15 min. The HRP enzyme labeling time was fixed at 1 min. Four-parameter logistic regression was used to fit the standard curve to data points taken for cytokine spiked-in 100% fetal bovine serum.





**Figure S7.** Standard curve of IL-6 ELISA for patient sample correlation. The ELISA kit was purchased from R&D Systems (Human IL-6 DuoSet ELISA DY206). The limit of detection was determined to be 12.03 pg/mL using the blank signal + 3 $\sigma$  (dotted line).

## MATLAB Training code for the convolutional neural network:

The training process of the in house developed dual-pathway CNN algorithm involves data set labeling and 2-step neural network training, which start from a 32×32 pixel locally-cropped images and are later applied to the 256×256 pixel large-sized images. The MATLAB training code is attached as follows, which includes file preparation (raw images and labels), neural network layer setup (downsampling and upsampling process), solver setup, class balance (for better accuracy), and post-processing (for quick testing). The detailed training method can be found in our previous publication<sup>1</sup>.

### File Preparation

```
clear;
close all force;
clc;

imageDir = fullfile('E:\bio research\rawimage'); %Provide the correct folder
path
labelDir = fullfile('E:\bio research\positive_label');
imds = imageDatastore(imageDir); %Contains all raw images

classNames = ["Qred","background"];
labelIDs = [1 0];
pxds = pixelLabelDatastore(labelDir,classNames,labelIDs); %Contains all labeled
images
```

### Layer setup

```
inputSize = [256 256 1]; %Input image size
numFilters = 10;
filterSize = 3;
numClasses = 2;
poolSize = 2;
layers = [
    imageInputLayer(inputSize) %Image Input layer
    convolution2dLayer(filterSize,numFilters,'Padding',1) %Downsampling network
    reluLayer()
    maxPooling2dLayer(poolSize,'Stride',2)
    convolution2dLayer(filterSize,numFilters,'Padding',1)
    reluLayer() %Downsampling end
    transposedConv2dLayer(4,numFilters,'Stride',2,'Cropping',1); %Upsampling
    reluLayer()
```

```

convolution2dLayer(1,numClasses);    %Pixel label layer
softmaxLayer()
pixelClassificationLayer()
]

```

## Solver Setup with class balance

```

opts = trainingOptions('sgdm', ...
    'InitialLearnRate',1e-3, ...
    'MaxEpochs',30, ...
    'MiniBatchSize',10, ...
    'Shuffle',"every-epoch", ...
    'Plots','training-progress');

trainingData = pixelLabelImageDatastore(imds,pxds);
tbl = countEachLabel(trainingData)
totalNumberOfPixels = sum(tbl.PixelCount);
frequency = tbl.PixelCount / totalNumberOfPixels;
inverseFrequency = 1./frequency;
layers(end) =
pixelClassificationLayer('Classes',tbl.Name,'ClassWeights',inverseFrequency); %Modify the last layer!
net = trainNetwork(trainingData,layers,opts);

```

## Post Processing

```

testImage = imread('E:\biodata\QRed00001.JPG');
Img_Qred = imcrop(testImage,[1101 301 3799 3399]);
Img_Qred_red = squeeze(Img_Qred(:,:,1));
figure,imshow(Img_Qred_red)
Mask_raw = semanticseg(testImage,net,'ExecutionEnvironment',"cpu");
Mask_Qred=Mask_raw=="Qred";
Mask_Qred = imfill(Mask_Qred, 'holes');
processedImage = labeloverlay(testImage,Mask_Qred);
figure,imshow(processedImage)

```