

APPENDIX

CODING TRAINING CNN

```
567. let cnn;
568. let jerukbusuk = [];
569. let jeruksegar = [];
570. let daun = [];
571.
572.
573. console.log('ml5 version:', ml5.version);
574. const IMAGE_WIDTH = 64;
575. const IMAGE_HEIGHT = 64;
576. const IMAGE_CHANNELS = 4;
577.
```

LOAD TRAINING IMAGE

```
578. function preload(){
579.   let j = 817;
580.   let a = 9;
581.   for(let i = 0; i < 10 ; i++){
582.     jerukbusuk[i] =
loadImage(`dataset/800x600/train/rottenoranges/jerukbusuk(${j}).png`);
583.     jeruksegar[i] =
loadImage(`dataset/800x600/train/freshoranges/jeruksegar(${j}).png`);
584.     j++;
585.     a++;
586.   }
587. }
588.
589.
```

SETUP CNN

```
590. function setup(){
591.   createCanvas(400,400);
592.   let options = {
593.     task: 'imageClassification',
594.     inputs: [IMAGE_WIDTH, IMAGE_HEIGHT, IMAGE_CHANNELS],
595.     debug: true,
596.   };
597.   cnn = ml5.neuralNetwork(options);
```

LOAD TRAINING IMAGE INTO CNN

```
598.   for(let f = 0; f < jerukbusuk.length; f++){
```

```
599.     let i = 0;
600.     cnn.addData({image : jerukbusuk[i]},{label : "Busuk"});
601.     cnn.addData({image : jeruksegar[i]},{label : "Segar"});
602.     i++;
603.     }
```

START TRAINING

```
604.     console.log('add data selesai');
605.     cnn.normalizeData();
606.     console.log('normalizeData jalan');
607.     stopWatch();
608.     cnn.train({epochs : 50},trainingselesai);
609. }
```

SAVE TRAINING RESULT

```
610. function trainingselesai(){
611.     console.log('training selesai');
612.     cnn.save();
613. }
```

TRAINING CNN HTML PAGE

```
614. <html>
615. <head>
616.     <meta charset="UTF-8">
```

CONNECT TO ML5 AND P5 LIBRARY

```
617.     <!-- p5 -->
618.     <script
        src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.2.0/p5.min.js"></
        script>
619.     <script
        src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.2.0/addons/p5.sound.
        min.js"></script>
620.     <!-- ml5 -->
621.     <script
        src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
622. </head>
623. <body>
624.     asdsadasdasd
625.     <div id="time">
626.         <span class="digit" id="hr">
627.             00</span>
628.         <span class="txt">Hr</span>
629.         <span class="digit" id="min">
630.             00</span>
```

```
631. <span class="txt">Min</span>
632. <span class="digit" id="sec">
633.   00</span>
634. <span class="txt">Sec</span>
635. <span class="digit" id="count">
636.   00</span>
637. </div>
```

CONNECT HTML TO CNN TRAINING CODE

```
638. <script src="cnn.js"></script>
639. </body>
640. </html>
```

CODING OBJECT DETECTION

```
641. let cnn;//repeat use variable
642. let video;
643. let testing = [];
644. let hasildiv;
645. let deteksi = [];
646. let roi;
647. let recording = [] ;
648. let recordingx = [];
649. let recordingy = [];
650. let f = 0;
651. let z = 0;
652. let recordroi = [];
653. let c = 0;
654. let nama ;
655. let width = 2;
656. let height = 2;
657. let k = 0 ;
658. let recordnama = [];
659. let recordlog = [] ;
660. let recordx = [];
661. let recordy = [];
662. let recordx2 = [];
663. let recordy2 = [];
664. let recordprediksi = [];
665. let prediksi = [];
666. let label = [];
667. let prediksi2 = [];
668. let label2 = [];
669. let akurasi = 0;
```

```

670. let x = [];
671. let y = [];
672. let w = [];
673. let h = [];
674. let w2 = [];
675. let h2 = [];
676. let recordw = [];
677. let recordw2 = [];
678. let recordh = [];
679. let recordh2 = [];
680. let hasil = [];
681. let b = 0;//repeat use variable
682. let koordinatfinal = [];//repeat use variable
683. let groundtruth = [];//repeat use variable
684. let prediksifinal = [];//repeat use variable
685. let tabeliou = [];//repeat use variable
686. let filename;
687. let averageprecision = [];
688. let predictionlength = 0; // repeatusevariable
689. let groundtruthname ;
690. let groundtruthlength = 0; // repeat use variable
691. let kelas = ['Segar','Busuk'];
692. let map = [];

```

GROUND TRUTH DATA

```

693. groundtruth = [
694.   ['jeruksegar(1).png',
695.    [
696.     ['Segar','100%',58,99,455,439]
697.    ]
698.  ],
699.  ['jeruksegar(2).png',
700.   [
701.    ['Segar','100%',25,31,354,306]
702.   ]
703.  ],
704.  ],
705.  ['jeruksegar(3).png',
706.   [
707.    ['Segar','100%',40,47,407,383]
708.   ]
709.  ],
710.  ],
711.  ['jeruksegar(22).png',
712.   [

```

```
713.      [
714.          ['Segar', '100%', 89, 31, 291, 121] ,
715.          ['Segar', '100%', 46, 122, 286, 355] ,
716.          ['Segar', '100%', 287, 63, 501, 278]
717.      ]
718.      ]
719.      ],
720.      ['jeruksegar (16) .png' ,
721.      [
722.          ['Segar', '100%', 1, 17, 563, 446]
723.      ]
724.      ]
725.      ],
726.      ['jeruksegar (243) .png' ,
727.      [
728.          ['Segar', '100%', 8, 21, 171, 147] ,
729.          ['Segar', '100%', 9, 147, 175, 280] ,
730.          ['Segar', '100%', 5, 280, 171, 416]
731.      ]
732.      ]
733.      ],
734.      ['jeruksegar (226) .png' ,
735.      [
736.          ['Segar', '100%', 28, 21, 314, 296]
737.      ]
738.      ]
739.      ],
740.      ['jeruksegar (240) .png' ,
741.      [
742.          ['Segar', '100%', 25, 13, 349, 333] ,
743.          ['Segar', '100%', 175, 104, 539, 432] ,
744.          ['Segar', '100%', 382, 41, 680, 354]
745.      ]
746.      ]
747.      ],
748.      ['jeruksegar (246) .png' ,
749.      [
750.          ['Segar', '100%', 11, 8, 397, 341]
751.      ]
752.      ]
753.      ],
754.      ['jeruksegar (248) .png' ,
755.      [
756.          ['Segar', '100%', 25, 21, 272, 269] ,
757.          ['Segar', '100%', 157, 79, 439, 368]
```

```
758.
759.     ]
760. ],
761. ['jerukbusuk(1) .png' ,
762.     [
763.         ['Segar' , '100%' , 43,32,365,198] ,
764.         ['Segar' , '100%' , 82,202,237,347] ,
765.         ['Segar' , '100%' , 242,202,324,288] ,
766.         ['Busuk' , '100%' , 372,124,419,196] ,
767.         ['Busuk' , '100%' , 329,203,419,288] ,
768.         ['Busuk' , '100%' , 243,294,389,346] ,
769.         ['Busuk' , '100%' , 165,349,371,418]
770.     ]
771.
772. ]
773. ],
774. ['jerukbusuk(2) .png' ,
775.     [
776.         ['Busuk' , '100%' , 26,16,290,277]
777.     ]
778. ],
779.
780. ['jerukbusuk(3) .png' ,
781.     [
782.         ['Busuk' , '100%' , 113,49,313,196] ,
783.         ['Segar' , '100%' , 33,57,107,197] ,
784.         ['Segar' , '100%' , 249,3,456,47] ,
785.         ['Segar' , '100%' , 318,50,515,201] ,
786.         ['Segar' , '100%' , 29,201,502,356]
787.     ]
788. ]
789. ],
790.
791. ['jerukbusuk(6) .png' ,
792.     [
793.         ['Busuk' , '100%' , 25,14,203,230]
794.     ]
795. ],
796. ['jerukbusuk(8) .png' ,
797.     [
798.         ['Busuk' , '100%' , 29,34,230,185] ,
799.         ['Busuk' , '100%' , 231,54,449,247] ,
800.         ['Busuk' , '100%' , 29,133,259,356]
801.     ]
802. ],
```

```

803.     ['jerukbusuk (222) .png' ,
804.     [
805.         ['Busuk' , '100%' , 52 , 1 , 310 , 247]
806.     ]
807. ],
808. ['jerukbusuk (229) .png' ,
809.     [
810.         ['Segar' , '100%' , 24 , 9 , 310 , 247] ,
811.         ['Busuk' , '100%' , 23 , 128 , 252 , 252]
812.     ]
813. ],
814. ['jerukbusuk (270) .png' ,
815.     [
816.         ['Busuk' , '100%' , 70 , 14 , 343 , 301]
817.     ]
818. ],
819. ['jerukbusuk (248) .png' ,
820.     [
821.         ['Segar' , '100%' , 31 , 273 , 397 , 410] ,
822.         ['Busuk' , '100%' , 30 , 15 , 409 , 268]
823.     ]
824. ],
825. ['jerukbusuk (233) .png' ,
826.     [
827.         ['Busuk' , '100%' , 51 , 29 , 265 , 242]
828.     ]
829. ]
830. ]
831. ];
832. console.log("ml5 version:" , ml5.version);
833. // opencv js image processign script
834. let imgElement = document.getElementById('imageSrc');
835. let inputElement = document.getElementById('fileInput');
836. inputElement.addEventListener('change' , (e) => {
837.     imgElement.src = URL.createObjectURL(e.target.files[0]);
838. }, false);

```

READ INPUTTED IMAGE

```

839. imgElement.onload = function() {
840.     refresh();
841.     let src = cv.imread(imgElement);
842.     let dst = new cv.Mat();
843.     let imgwidth = imgElement.width / 2;
844.     let imgheight = imgElement.height / 2;
845.     let roix1 = 0;

```

```

846.     let roix2 = 0;
847.     let roiy1 = imgwidth;
848.     let roiy2 = imgheight;
849.     let i = 0; // rubah i menjadi 1 untuk merecord coordinat
           menggunakan titik paling awal
850.     x[i] = roix1;
851.     w[i] = roix1 + imgwidth;
852.     y[i] = roix2;
853.     h[i] = roix2 + imgheight;
854.     console.log("nilai awal" , roix1);

```

PROCESS INPUTTED IMAGE

```

855.     while(roix1 != imgElement.width  &&  roix2  !=
imgElement.height){
856.         let rect = new cv.Rect(roix1,roix2,roiy1,roiy2);
857.         dst = src.roi(rect);
858.         cv.imshow('canvasOutput', dst);
859.         resizeimg();
860.         i++;
861.         klasifikasi();
862.
863.         if(roix1 != imgElement.width){
864.             roix1 = roix1 + imgwidth;
865.
866.         }
867.         x[i] = roix1;
868.         w[i] = roix1 + imgwidth;
869.
870.         if(roix1 == imgElement.width  &&  roix2  <=
imgElement.height){
871.             roix1 = imgwidth;
872.             roix2 = roix2 + imgheight;
873.
874.         }
875.         y[i] = roix2;
876.         h[i] = roix2 + imgheight;
877.
878.         f++;
879.     }
880.     console.log("roix1 = ", x);
881.     console.log("roix2 = ", y);
882.     console.log("w = ", w);
883.     console.log("h = ", h);
884.     console.log(i);
885.

```



```

886.
887.     src.delete();
888.     dst.delete();
889.
890.
891. };
892.
893. var Module = {
894.                                     //
      https://emscripten.org/docs/api_reference/module.html#Module.onRun
      timeInitialized
895.     onRuntimeInitialized() {
896.         document.getElementById('status').innerHTML = 'OpenCV.js
      is ready.';
897.     }
898. };

```

PROCESS DOWNSIZED INPUTED IMAGE

```

899. function pyramid(){
900.     let src = cv.imread(imgElement);
901.     let dst = new cv.Mat();
902.     x = [];
903.     y = [];
904.     recordnama = []
905.     recordprediksi = []
906.
907.     cv.pyrDown(src, dst, new cv.Size(0, 0), cv.BORDER_DEFAULT);
908.     cv.imshow('canvasOutput', dst);
909.     src.delete(); dst.delete();
910.     src = cv.imread('canvasOutput');
911.     dst = new cv.Mat();
912.     let imgwidth = src.size().width / width;
913.     let imgheight = src.size().height / height;
914.     console.log('pyramid height ' , imgheight);
915.     let roix1 =0;
916.     let roix2 = 0;
917.     let roiy1 = roix1 + imgwidth;
918.     let roiy2 = roix2 + imgheight;
919.     let save = 0;
920.     console.log('roix1',roix1)
921.     let i = 0; // rubah i menjadi 1 untuk merecord koordinat
      menggunakan titik paling awal
922.     x[i] = 0;
923.     w2[i] = x[i] + imgwidth;
924.     y[i] = 0;
925.     h2[i] = y[i] + imgheight;

```

```

926.     console.log("nilai awal" , roix1);
927.     // You can try more different parameters
928.     while(roix1  !=  src.size().width  &&  roix2  !=
src.size().height) {
929.         let rect = new cv.Rect(roix1,roix2,roiy1,roiy2);
930.         dst = src.roi(rect);
931.         cv.imshow('canvasOutput', dst);
932.         i++;
933.         resizeimg();
934.
935.
936.         klasifikasi();
937.
938.         if(roix1 != src.size().width){
939.             roix1 = roix1 + imgwidth;
940.
941.         }
942.         x[i] = roix1;
943.         w2[i] = x[i] + imgwidth;
944.         // if(roix1 > src.size().width){
945.         //     roix1 = src.size().width;
946.
947.         // }
948.         if(roix1 == src.size().width  &&  roix2  <=
src.size().height) {
949.             roix1 = imgwidth;
950.             roix2 = roix2 + imgheight;
951.
952.         }
953.         y[i] = roix2;
954.         h2[i] = y[i] + imgheight;
955.
956.         f++;
957.     }
958.     console.log("roix1 = ", x);
959.     console.log("roix2 = ", y);
960.     console.log("w = ", w);
961.     console.log("h = ", h);
962.     console.log(i);
963.     src.delete();
964.     dst.delete();
965. }

```

RESIZE INPUTTED IMAGE

```

966. function resizeimg(){

```

```

967.     let srca = cv.imread('canvasOutput');
968.     let dsta = new cv.Mat();
969.     let dsize = new cv.Size(200, 100);
970.     cv.resize(srca, dsta, dsize, 0, 0, cv.INTER_AREA);
971.     cv.imshow('canvasOutput', dsta);
972.     srca.delete(); dsta.delete();
973.
974.
975.
976.     }

```

SETUP CNN

```

977.     function setup() {
978.         createCanvas(400, 400);
979.
980.         let options = {
981.             task: "imageClassification",
982.             input: [200, 100, 4],
983.         };
984.         cnn = ml5.neuralNetwork(options);
985.         const modelDetails = {
986.             model: "model/200x100/model.json",
987.             metadata: "model/200x100/model_meta.json",
988.             weights: "model/200x100/model.weights.bin",
989.         };
990.         hasildiv = createDiv("Loading model...");
991.
992.         cnn.load(modelDetails, modelReady);
993.
994.     }
995.
996.
997.     function modelReady() {
998.         console.log("model loaded!");
999.         hasildiv.html("Model ready");
1000.
1001.     }

```

IMAGE CLASSIFICATION

```

1002.     function klasifikasi() {
1003.
1004.         roi = document.getElementById("canvasOutput");
1005.         recordroi[f] = roi;
1006.         cnn.classify({ image: roi }, handleResults);
1007.

```

```

1008.
1009. }
1010.
1011. function handleResults(error, result) {
1012.     if (error) {
1013.         console.error(error);
1014.         return;
1015.     }
1016.
1017.     deteksi = result;
1018.     nama = result[0].label;
1019.     akurasi = nf(100 * result[0].confidence, 2, 0);
1020.     hasildiv.html(`${nama} ${akurasi}%`);
1021.     console.log(deteksi);
1022.     // recording[f] = roi;
1023.     recordnama[z] = nama;
1024.     recordprediksi[z] = akurasi;
1025.     console.log('recordnama length = ', recordnama);
1026.     console.log('recording prediksi= ', recordprediksi);
1027.     // console.log('record roi koordinat', recordx);
1028.     z++;
1029.     if ( z == f){
1030.         record();
1031.     }
1032.
1033.     // klasifikasi();
1034. }

```

RECORD PROCESSED DATA

```

1035. function record(){
1036.     let n = 0;
1037.
1038.     //penyimpanan data hasil klasifikasi data asli
1039.     if(c == 0 ){
1040.         for(i = 0 ; i < x.length ; i ++){
1041.             //filter data yang memiliki nilai prediksi > 60
1042.             if (recordprediksi[i] > 20){
1043.                 recordx[n] = Math.round(x[i]);
1044.                 recordy[n] = Math.round(y[i]);
1045.                 label[n] = recordnama[i];
1046.                 prediksi[n] = recordprediksi[i];
1047.                 recordw[n] = w[i];
1048.                 recordh[n] = h[i];
1049.                 n++;
1050.             }

```

```

1051.     }
1052.     console.log('mulai pyramid');
1053.     z = 0;
1054.     f = 0;
1055.     recordnama=[];
1056.     recordprediksi = [];
1057.
1058.     x = [];
1059.     y = [];
1060.     c++;
1061.     pyramid();
1062.     }else{
1063.         n = 0;
1064.         //penyimpanan data hasil klasifikasi data pyramid
1065.         for(i = 0 ; i < x.length ; i ++){
1066.             //filter data yang memiliki nilai prediksi > 60
1067.             if (recordprediksi[i] > 20 ){
1068.                 recordx2[k] = Math.round(x[i]);
1069.                 recordy2[k] = Math.round(y[i]);
1070.                 label2[k] = recordnama[i];
1071.                 prediksi2[k] = recordprediksi[i];
1072.                 recordw2[k] = w2[i];
1073.                 recordh2[k] = h2[i];
1074.                 k++;
1075.             }
1076.         }
1077.         console.log('recordnama label = ' , label);
1078.         console.log('record prediksi= ' , prediksi);
1079.         console.log('record x koordinat' , recordx);
1080.         console.log('record y koordinat' , recordy);
1081.         console.log('record prediksi2 = ',prediksi2);
1082.         console.log('record x2 koordinat' , recordx2);
1083.         console.log('record y2 koordinat' , recordy2);
1084.         console.log('record b ',b);
1085.         nms ();
1086.     }
1087.
1088. }

```

APPLY NMS FUNCTION

```

1089. function nms(){
1090.     //mencari file name
1091.     let tempgroundtruthlength;
1092.     var inp = document.getElementById('fileInput');
1093.     for (i = 0; i < inp.files.length; i++) {

```

```

1094.     let file = inp.files[i];
1095.         filename = file['name'];
1096.
1097.     }
1098.     //mencari panjang groundtruth yang mempunyai nama sama dengan
        filename
1099.     for ( let z = 0 ; z < groundtruth.length ; z++){
1100.         groundtruthname = groundtruth[z][0];
1101.         if(filename == groundtruthname){
1102.             console.log('ground truth full',groundtruth[z]);
1103.             for(let f = 0 ; f < groundtruth[z][1].length; f++ ){
1104.                 groundtruthlabel = groundtruth[z][1][f][0];
1105.
1106.                 tempgroundtruthlength = groundtruth[z][1].length;
1107.             }
1108.             if(b > 0){
1109.                 groundtruthlength = groundtruthlength +
        tempgroundtruthlength ;
1110.             }else{
1111.                 groundtruthlength = tempgroundtruthlength ;
1112.             }
1113.         }
1114.     }
1115.
1116.
1117.     prediksi = prediksi.map(str => {
1118.         return Number(str);
1119.     });
1120.     prediksi2 = prediksi2.map(str => {
1121.         return Number(str);
1122.     });
1123.     let boxes = [];
1124.     let recordxfinal = recordx.concat(recordx2);
1125.     let recordyfinal = recordy.concat(recordy2);
1126.     let recordwfinal = recordw.concat(recordw2);
1127.     let recordhfinal = recordh.concat(recordh2);
1128.     let labelfinal = label.concat(label2);
1129.
1130.     const prediksinomor = prediksi.concat(prediksi2);
1131.     for ( i = 0 ; i < recordxfinal.length ; i ++){
1132.         boxes[i] =
        [recordyfinal[i],recordxfinal[i],recordhfinal[i],recordwfinal[i]];
1133.         // prediksinomor[i] = [prediksi[i],prediksi2[i]];
1134.
1135.     }
1136.     let text = prediksinomor.map(String);

```

```

1137.     //merubah data koordinat menjadi tensor agar dapat
           digunakan di nms
1138.     const shape = [boxes.length ,4];
1139.     const boxes1 = tf.tensor2d(boxes,shape);
1140.     console.log('isi boxes' , boxes1.arraySync());
1141.     console.log('panjang boxes',typeof recordxfinal);
1142.     console.log('label final' , labelfinal);
1143.     console.log('Score final',prediksinomor);
1144.     // Calling image.nonMaxSuppression() method and
1145.     // Printing output
1146.     //menghilangkan kotak prediksi yg overlap
1147.     const nms = tf.image.nonMaxSuppression(boxes1,
           prediksinomor,tempgroundtruthlength + 1 ,0.5);
1148.
1149.     let mat = cv.imread(imgElement);
1150.     hasil = nms.arraySync();
1151.     console.log('hasil nms',hasil);
1152.     console.log('isi text',text);
1153.
1154.     let font = cv.FONT_HERSHEY_SIMPLEX
1155.     //visualisasi hasil prediksi
1156.     for(let f = 0 ; f < hasil.length ; f++){
1157.         let point1 = new cv.Point(recordxfinal[hasil[f]],
           recordyfinal[hasil[f]]);
1158.         let point2 = new cv.Point(recordwfinal[hasil[f]],
           recordhfina[hasil[f]]);
1159.         let point3 = new cv.Point(recordxfinal[hasil[f]],
           recordyfinal[hasil[f]] - 10 );
1160.         let point4 = new cv.Point(recordwfinal[hasil[f]] + 10,
           recordhfina[hasil[f]]);
1161.         let point5 = new cv.Point(recordwfinal[hasil[f]] - 20,
           recordhfina[hasil[f]] - 20);
1162.         let z = f.toString();
1163.         console.log('hasil[f]',hasil[f]);
1164.         console.log('recordxfinal[f]',recordxfinal[hasil[f]]);
1165.         console.log('recordyfinal[f]',recordyfinal[hasil[f]]);
1166.         console.log('recordwfinal[f]',recordwfinal[hasil[f]]);
1167.         console.log('recordhfina[f]',recordhfina[hasil[f]]);
1168.         cv.putText(mat,labelfinal[hasil[f]],point3, font, 0.5,
           [0,0,255,0],1,cv.LINE_AA);
1169.         cv.putText(mat,text[hasil[f]],point4, font, 0.5,
           [0,0,255,0],1,cv.LINE_AA);
1170.         cv.putText(mat,z,point5, font, 0.5,
           [0,0,255,0],1,cv.LINE_AA);
1171.         cv.rectangle(mat, point1,point2, [0, 0, 30, 255 ], 3);
1172.     }
1173.     //visualisasi groundtruth

```



```

1174.   for ( let z = 0 ; z < groundtruth.length ; z++){
1175.       groundtruthname = groundtruth[z][0];
1176.       if(filename == groundtruthname){
1177.           for(let f = 0 ; f < groundtruth[z][1].length; f++ ){
1178.               let point1 =  new cv.Point(groundtruth[z][1][f]
1179. [2],groundtruth[z][1][f][3]);
1180.               let point2 = new cv.Point(groundtruth[z][1][f]
1181. [4],groundtruth[z][1][f][5]);
1182.               cv.rectangle(mat, point1,point2, [0, 0, 255, 255 ],
1183. 3);
1184.           }
1185.       }
1186.   }
1187.   console.log('b',b);
1188.   cv.imshow('canvasOutput', mat);
1189.   mat.delete();
1190.   let koordinatprediksi = [];
1191.   var inp = document.getElementById('fileInput');
1192.   // Access and handle the files
1193.   f = 0;
1194.   //memasukan semua data prediksi ke dalam 1 array
1195.   for (i = 0; i < inp.files.length; i++) {
1196.       let file = inp.files[i];
1197.       for (let k = 0 ; k < hasil.length ; k++){
1198.           koordinatprediksi[k] =
1199. [file['name'],labelfinal[hasil[f]],text[hasil[f]],
1200. recordxfinal[hasil[f]], recordyfinal[hasil[f]],
1201. recordwfinal[hasil[f]], recordhfinal[hasil[f]]];
1202.           filename = file['name'];
1203.           f++;
1204.       }
1205.   }
1206.   // do things with file
1207.   koordinatfinal[b] = koordinatprediksi;
1208.
1209.
1210.
1211.   console.log('ground truth label',groundtruth[0][1]);
1212.   console.log('ground truth full',groundtruth);
1213.   console.log('koordinat prediksi',koordinatprediksi);
1214.   console.log('koordinat final nama file',koordinatfinal[0]
[0][0]);

```



```

1215.     console.log('koordinat final full',koordinatfinal);
1216.     console.log('b',b);
1217.     for ( i = 0 ; i < groundtruth.length ; i++){
1218.         if(filename == groundtruth[i][0]){
1219.             iou();
1220.         }
1221.     }
1222.
1223.     // b++;
1224.     // mean();
1225.
1226. }
1227.

```

REFRESH GLOBAL VARIABLE

```

1228. function refresh(){
1229.     cnn;//repeat use variable
1230.     video;
1231.     testing = [];
1232.     hasildiv;
1233.     deteksi = [];
1234.     // dst ;
1235.     roi;
1236.     recording = [] ;
1237.     recordingx = [] ;
1238.     recordingy = [] ;
1239.     f = 0;
1240.     z = 0;
1241.     recordroi = [] ;
1242.     c = 0;
1243.     nama ;
1244.     recordnama = [] ;
1245.     recordlog = [] ;
1246.     recordx = [] ;
1247.     recordy = [] ;
1248.     recordx2 = [] ;
1249.     recordy2 = [] ;
1250.     recordprediksi = [] ;
1251.     prediksi = [] ;
1252.     label = [] ;
1253.     prediksi2 = [] ;
1254.     label2 = [] ;
1255.     akurasi = 0;
1256.     x = [] ;
1257.     y = [] ;

```

```

1258.     w = [];
1259.     h = [];
1260.     w2 = [];
1261.     h2 = [];
1262.     recordw = [];
1263.     recordw2 = [];
1264.     recordh = [];
1265.     recordh2 = [];
1266.     hasil = [];
1267.     filename;
1268.     groundtruthname ;
1269.
1270. }
1271.

```

CALCULATING IOU

```

1272. function iou(){
1273.     let xa = [];
1274.     let xb = [];
1275.     let ya = [];
1276.     let yb = [];
1277.     let iou = [];
1278.     let labeliou = [];
1279.     let tp = [];
1280.     let fp = [];
1281.     let fn = [];
1282.     let tptotal = [];
1283.     let fptotal = [];
1284.     let fntotal = [];
1285.     let interArea;
1286.     let boxAArea;
1287.     let boxBArea;
1288.     let threshold = 0.3;
1289.     let z = 0;
1290.     let n = 0 ;
1291.     let groundtruthlabel = groundtruth[z][1];
1292.     let groundtruthname = groundtruth[z][0];
1293.     let precision = [] ;
1294.     let recall = [] ;
1295.     let totaliou = [];
1296.     let precisiontotal = [];
1297.     let recalltotal = [];
1298.     let m = 1;
1299.
1300.     let confidence = [] ;

```

```

1301.
1302. // console.log('ground truth full',groundtruth);
1303. //hitung iou
1304. for ( let z = 0 ; z < groundtruth.length ; z++){
1305.     groundtruthname = groundtruth[z][0];
1306.
1307.     if(filename == groundtruthname){
1308.         console.log('ground truth full',groundtruth[z]);
1309.         for(let i = 0 ; i < koordinatfinal[b].length ; i +
1310. +) {
1311.             for(let f = 0 ; f < groundtruth[z][1].length; f++
1312. ) {
1313.                 groundtruthlabel = groundtruth[z][1][f][0];
1314.                 console.log('groundtruth[z]
1315. [1].length',groundtruth[z][1].length);
1316.                 xa = max([groundtruth[z][1][f][2] ,
1317. koordinatfinal[b][i][3]]);
1318.                 ya = max([groundtruth[z][1][f][3] ,
1319. koordinatfinal[b][i][4]]);
1320.                 xb = min([groundtruth[z][1][f][4] ,
1321. koordinatfinal[b][i][5]]);
1322.                 yb = min([groundtruth[z][1][f][5] ,
1323. koordinatfinal[b][i][6]]);
1324.                 interArea = max(0, xb - xa +1 ) * max(0, yb -
1325. ya +1);
1326.                 boxAArea = (groundtruth[z][1][f][4] -
1327. groundtruth[z][1][f][2] +1) *
1328. (groundtruth[z][1][f][5] - groundtruth[z][1][f]
1329. [3] +1);
1330.                 boxBArea = (koordinatfinal[b][i][5] -
1331. koordinatfinal[b][i][3] +1) *
1332. (koordinatfinal[b][i][6] - koordinatfinal[b]
1333. [i][4]+1);
1334.                 labeliou[i] = koordinatfinal[b][i][1];
1335.                 iou[f] = interArea / float(boxAArea +
1336. boxBArea - interArea);
1337.                 console.log('iou',iou);
1338.             }
1339.             //cari tp fp dan fn
1340.             totaliou[i] = max(iou);
1341.             if(totaliou[i] > threshold && labeliou[i] ==
1342. groundtruthlabel) {
1343.                 tp.push([filename,i,labeliou[i],totaliou[i]]);
1344.             }else if(totaliou[i] > 0 && totaliou[i]
1345. <threshold)
1346.             {
1347.                 fp.push ([filename,i,labeliou[i],totaliou[i]]);

```

```

1334.
1335.         }
1336.     else{
1337.         fn.push ([filename,i,labeliou[i],totaliou[i]]);
1338.
1339.     }
1340.
1341.     tptotal[i] = tp.length;
1342.     fptotal[i] = fp.length;
1343.     fntotal[i] = fn.length;
1344.     // if(b> 0){
1345.
1346.         // }else{
1347.
1348.         // }
1349.     predictionlength = predictionlength + m;
1350.     // confidence[i] = koordinatfinal[b][i][2];
1351.     precision[i] = tp.length / (tp.length +
fp.length);
1352.     precision[i] = precision[i] || 0;
1353.     recall[i] = tp.length / (tp.length + fn.length);
1354.     recall[i] = recall[i] || 0;
1355.     precisiontotal[i] = tp.length / predictionlength;
1356.     precisiontotal[i] = precisiontotal[i] || 0;
1357.     recalltotal[i] = tp.length /groundtruthlength;
1358.     recalltotal[i] = recalltotal[i] || 0;
1359.     console.log('precisiontotal[i]',precisiontotal[i]
)
)
1360.     console.log('totaliou.length',totaliou.length)
1361.     console.log('tp.length',tp.length)
1362.     console.log(' predictionlength',
predictionlength)
1363.     // m++;
1364.     // iou = [];
1365.     }
1366. }
1367. }
1368.     tabeliou[b] =
[labeliou,confidence,tptotal,fptotal,fntotal,precisiontotal,recall
total,iou];
1369.     prediksifinal.push( [precisiontotal,recalltotal,labelio
u]);
1370.     // precision.sort((a, b) => recall.indexOf(a) -
recall.indexOf(b));
1371.     // totalioulength = tp.length + fp.length + fn.length;
1372.

```

```

1373.
1374. console.log('tp', tp);
1375. console.log('fp', fp);
1376. console.log('fn', fn);
1377. console.log('prediksifinal', prediksifinal);
1378. console.log('totaltp', tptotal);
1379. console.log('totalfp', fptotal);
1380. console.log('totalfn', fntotal);
1381. console.log('prediction length', predictionlength);
1382. // console.log('totalioulength', totalioulength);
1383. console.log('precision', precision);
1384. console.log('recall', recall);
1385. console.log('groundtruth length', groundtruthlength);
1386. // console.log('totalioulength', totaliou);
1387. console.log('tabeliou', tabeliou);
1388. refresh();
1389. ap();
1390.
1391. }
1392.

```

CALCULATING AP AND MAP

```

1393. function ap(){
1394.     let interaptotal = 0;
1395.     let ap = [];
1396.     let recalltotal = [];
1397.     let precisiontotal = [];
1398.     let labeltotal = [];
1399.     let templabel = [];
1400.     let interlabel = [];
1401.
1402.     let m = 0;
1403.     let n = 0;
1404.     let z = 0 ;
1405.     let temprecall =[];
1406.     let tempprecision = [];
1407.     let interrecall = [];
1408.     let interprecision = [];
1409.     let interap = [];
1410.     let interapkelas = [];
1411.
1412.
1413.
1414.
1415.     //menggabungkan data precision dan recall

```

```

1416.   for(let i = 0 ; i < prediksifinal.length ; i++){
1417.       if(i > 0 ){
1418.           precisiontotal = precisiontotal.concat(prediksifinal[i]
1419. [0]);
1420.           recalltotal = recalltotal.concat(prediksifinal[i][1]);
1421.           labeltotal = labeltotal.concat(prediksifinal[i][2]);
1422.       }else{
1423.           precisiontotal = prediksifinal[i][0];
1424.           recalltotal = prediksifinal[i][1];
1425.           labeltotal = prediksifinal[i][2];
1426.       }
1427.   }
1428.   console.log('prediksifinal ap',prediksifinal);
1429.   for ( let i = 0 ; i < precisiontotal.length ; i ++){
1430.       if(temprecall.includes(recalltotal[i]) == false){
1431.           // temprecall[m] = recalltotal[i];
1432.           temprecall.push(recalltotal[i]);
1433.           tempprecision.push(precisiontotal[i]);
1434.           templabel.push(labeltotal[i]);
1435.           // console.log('i',i);
1436.       }else{
1437.           let f = temprecall.indexOf(recalltotal[i]);
1438.           let z = precisiontotal[f];
1439.           if(precisiontotal[i] > z){
1440.               precisiontotal[f] = precisiontotal[i];
1441.           }
1442.       }
1443.   }
1444.   }
1445.
1446.   for(let i = 0 ; i < tempprecision.length ; i++){
1447.       n = n + tempprecision[i];
1448.   }
1449.
1450.   ap = n / 11; //ap for 11 point interpolation for all class
1451.
1452.   for(let i = 0 ; i < tempprecision.length ; i++){
1453.       if(interprecision.includes(tempprecision[i]) == false){
1454.           interprecision[m] = tempprecision[i];
1455.           interrecall[m] = temprecall[i];
1456.           interlabel [m] = templabel[i];
1457.           console.log('interrecall m',interrecall[m]);
1458.           m++;
1459.

```

```

1460.     }else{
1461.         z= interrecall.length -1 ;
1462.
1463.         if(temprecall[i] > interrecall[z]){
1464.             interrecall[z] = temprecall[i];
1465.         }
1466.         console.log('i',i)
1467.
1468.     }
1469.     // console.log('interrecall m',interrecall[m])
1470. }
1471. for(let a = 0 ; a < kelas.length ; a++){
1472.     interaptotal = 0;
1473.     interap = [];
1474.     let k = 0
1475.     //calculating area under curve precision x recall
1476.     let l = 0;
1477.     for(let i = 0 ; i < interprecision.length ; i++){
1478.         if(interlabel[i] == kelas[a]){
1479.             interap[k] = interprecision[i] * (interrecall[i] -
1480. 1);
1481.             l = interrecall[i];
1482.             interap[k] = interap[k] || 0;
1483.             console.log('kelas a',kelas[a])
1484.             console.log('interlabel i',interlabel[i])
1485.             console.log('interprecision[i] ',interprecision[i] )
1486.             k++;
1487.         }
1488.     }
1489.     console.log('interap',interap)
1490.     //hitung hasil all point interpolation ap
1491.     for ( let i = 0; i < interap.length; i++){
1492.         interaptotal = interaptotal + interap[i];
1493.     }
1494.     interapkelas[a] = interaptotal || 0;
1495. }
1496. // let map;
1497. let aptotal = 0;
1498. for(let i = 0 ; i < interapkelas.length; i++){
1499.     aptotal = aptotal + interapkelas[i];
1500. }
1501. map[b] = aptotal / interapkelas.length;
1502. // console.log('prediksifinallength',prediksifinal.length);
1503. console.log('label total',labeltotal);

```

```

1504.     console.log('precision total' , precisiontotal);
1505.     console.log('recall total',recalltotal);
1506.     console.log('templabel', templabel);
1507.     console.log('temprecall', temprecall);
1508.     console.log('tempprecision', tempprecision);
1509.     console.log('interlabel', interlabel);
1510.     console.log('interprecision', interprecision);
1511.     console.log('interrecall', interrecall);
1512.     console.log('ap', aptotal);
1513.     console.log('interap', interap);
1514.     console.log('interaptotal', interaptotal);
1515.     console.log('interapkelas', interapkelas);
1516.     console.log('MAP', map);
1517.     b++;
1518. }

```

OBJECT DETECTION HTML PAGE

```

1519. <!DOCTYPE html>
1520. <html>
1521. <head>

```

CONNECT TO ML5 , AND P5 LIBRARY

```

1522.                                     <script
      src="https://cdn.jsdelivr.net/npm/p5@1.4.1/lib/p5.js"></script>
1523.     <script src="p5.sound.min.js"></script>
1524.                                     <script
      src="https://unpkg.com/ml5@latest/dist/ml5.min.js"></script>
1525.     <!-- <script src="ml5.min.js"></script> -->
1526.                                     <script
      src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.m
      in.js"></script>
1527.     <meta charset="utf-8">
1528.     <title>Hello OpenCV.js</title>
1529.     </head>
1530.     <body>
1531.     <h2>Hello OpenCV.js</h2>
1532.     <p id="status">OpenCV.js is loading...</p>
1533.     <div>

```

IMAGE INPUT AND OUTPUT AREA

```

1534.     <div class="inputoutput">
1535.         <img id="imageSrc" alt="No Image" />
1536.         <div class="caption">imageSrc <input type="file"
      id="fileInput" name="file" /></div>
1537.     </div>

```



```
1538. <div class="inputoutput">
1539.     <canvas id="canvasOutput"></canvas>
1540.     <div class="caption">canvasOutput</div>
1541. </div>
1542. </div>
```

CONNECT HTML TO OPENCV LIBRARY AND OBJECT

DETECTION CODE

```
1543. <script src = "sketch.js" type="text/javascript"></script>
1544. <script           async           src="opencv.js"
      type="text/javascript"></script>
1545. </body>
1546. </html>
```



PAPER NAME

TA-18.K1.0003.docx

WORD COUNT

9655 Words

CHARACTER COUNT

49283 Characters

PAGE COUNT

37 Pages

FILE SIZE

47.0KB

SUBMISSION DATE

Dec 12, 2022 9:11 AM GMT+7

REPORT DATE

Dec 12, 2022 9:12 AM GMT+7

● **7% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 2% Internet database
- 3% Publications database
- Crossref database
- Crossref Posted Content database
- 4% Submitted Works database

