
```

classdef GUI_Design_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                                matlab.ui.Figure
        UploadVideoButton                       matlab.ui.control.Button
        FrameSliderLabel                        matlab.ui.control.Label
        FrameSlider                             matlab.ui.control.Slider
        SaveProcessedVideoButton                matlab.ui.control.Button
        InstantaneousVelocityButton             matlab.ui.control.Button
        AverageVelocityButton                   matlab.ui.control.Button
        MaxSpreadRadiusButton                   matlab.ui.control.Button
        SaveAnalyticsButton                     matlab.ui.control.Button
        SatelliteDropletSpeedButton             matlab.ui.control.Button
        SatelliteDropletCountButton             matlab.ui.control.Button
        InstantaneousVelocityUnits              matlab.ui.control.EditField
        AverageVelocityUnits                    matlab.ui.control.EditField
        MaxSpreadUnits                          matlab.ui.control.EditField
        DropletSpeedUnits                       matlab.ui.control.EditField
        DropletCountUnits                       matlab.ui.control.EditField
        CurrentFrameEditFieldLabel              matlab.ui.control.Label
        CurrentFrameEditField                   matlab.ui.control.EditField
        InstantVelocityTxt                      matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        AverageVelocityTxt                      matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        MaxRadiusTxt                           matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        DropletSpeedTxt                        matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        DropletCountTxt                        matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        MicronspixelEditFieldLabel              matlab.ui.control.Label
        MicronspixelEditField                   matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        ContactAngleButton                      matlab.ui.control.Button
        ContactAngleUnits                       matlab.ui.control.EditField
        ContactAngleTxt                         matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        TotalSatelliteDropletsButton             matlab.ui.control.Button
        TotalSatelliteDropletsTxt               matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        MaxVelocityPrimaryDropletButton          matlab.ui.control.Button
        MaxVelocityPrimaryDropletTxt             matlab.ui.control.EditField
        matlab.ui.control.NumericEditField
        MaxVelocityPrimaryDropletUnits           matlab.ui.control.EditField
        TotalSatelliteDropletsUnits              matlab.ui.control.EditField
        OutlineColorDropDownLabel                matlab.ui.control.Label
        OutlineColorDropDown                     matlab.ui.control.DropDown
        SaveCurrentFrameButton                   matlab.ui.control.Button
        SpreadRadius                             matlab.ui.control.Button
        SpreadRadiusUnits                       matlab.ui.control.EditField
    end
end

```

```

        SpreadRadiusTxt
matlab.ui.control.NumericEditField
        ContactAnglesOffButton          matlab.ui.control.StateButton
        JetVelocity                      matlab.ui.control.Button
        JetVelocityUnits                 matlab.ui.control.EditField
        JetVelocityTxt
matlab.ui.control.NumericEditField
        JetTipPosition                  matlab.ui.control.Button
        JetTipPositionUnits              matlab.ui.control.EditField
        JetTipPositionTxt
matlab.ui.control.NumericEditField
        JetDiameter                     matlab.ui.control.Button
        JetDiameterUnits                 matlab.ui.control.EditField
        JetDiameterTxt
matlab.ui.control.NumericEditField
        FramesecLabel                   matlab.ui.control.Label
        FramesecEditField
matlab.ui.control.NumericEditField
        UIAxes                          matlab.ui.control.UIAxes
        UIAxes2                          matlab.ui.control.UIAxes
end

properties (Access = private)
    R % total size and frame count of original video
    B % matrix of processed border video frames with the floor
removed
    Ly % matrix of original video frames with a yellow border
    Lr % matrix of original video frames with a red border
    Lg % matrix of original video frames with a green border
    Lb % matrix of original video frames with a blue border
    cay % matrix of original video frames with yellow contact
angles
    car % matrix of original video frames with yellow contact
angles
    cag % matrix of original video frames with yellow contact
angles
    cab % matrix of original video frames with yellow contact
angles
    DialogApp % Dialog box app
    floorheight % pixel height of floor
    floorangle % angle of floor
    mspreadl % max spread radius left
    mspreadr % max spread radius right
    spreadl % spread radius left
    spreadr % spread radius right
    micron_to_pxl % pixel to micron ratio
    frame_to_s % frame to second ratio
    cangle % contact angle (both left and right) for the matrix
    cpoints % contact position for the contact angles printing
function
    fvelocity %fall velocity of droplet x and y directions
    impactdata % includes [speed,frame#] for impact of primary
droplet

```

```

        totalobjects % a count of the number of separate objects in
frame
        ctrmsprd % center found from maxspread function
        viddisplay % the current video frames to display
        jetv % jet velocity
        jetpos % height of the tip of the jet
        jetd % jet diameter
        mxvindex % frame of the max velocity
    end

    properties (Access = public)
        M % matrix of original video frames
    end

    methods (Access = public)
        function updatefloor(app, h, t)
            % Store inputs as properties
            app.floorheight = h;
            app.floorangle = t;
        end
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: UploadVideoButton
        function UploadVideoButtonPushed(app, event)
            [vidname,vidpath] = uigetfile('*.avi'); % this asks the
user for a file input
            if vidname == 0
                figure(app.UIFigure); % this ensures that the app does
not go to the background when uigetfile is called
                return; % this is a simple catch to stop the program
from displaying an error if the user presses cancel
            end
            figure(app.UIFigure); % this ensures that the app does not
go to the background when uigetfile is called
            fullname = fullfile(vidpath,vidname); % this concatenates
the path and file name of the input file
            [app.M,app.R] = video2frame(fullname);
            imshow(app.M(:,:,1), 'Parent', app.UIAxes); % this
displays the input file in the original video axes
            totalframes = app.R(4); % this sets totalframes to the
number of frames in the uploaded video
            app.FrameSlider.Limits = [1 totalframes]; % this sets the
slider limits to 1 to the total frames of the video
            app.FrameSlider.MajorTicks = [1:30:totalframes
totalframes]; % this sets the slider major ticks to 1 to the total
frames
            % of the video in increments of 20
            app.FrameSlider.MinorTicks = [1:5:totalframes]; % this
sets the slider minor ticks to 1 to the total frames
            % of the video in increments of 5

```

```

        app.DialogApp = floorselectdlgbox(app,app.M);
        waitfor(app.DialogApp);
        B_withfloor = borders(app.M); % this processes the video
to get a B&W border video
        app.B =
        floorremove(B_withfloor,app.floorheight,app.floorangle); % this
processes the border video to remove the floor
            rotatedsource = convertSource(app.M,app.floorangle); %
this rotates the source image for use in maskOverlay function
            omask = outlineMask(app.B); % this creates an outline mask
from the border matrix
            app.Ly = maskOverlay(rotatedsource,omask,5,[255,255,0]); %
this creates a matrix of frames with the original video
            % with a yellow line around the border
            app.Lr = maskOverlay(rotatedsource,omask,5,[255,0,0]); %
this creates a matrix of frames with the original video
            % with a red line around the border
            app.Lg = maskOverlay(rotatedsource,omask,5,[0,255,0]); %
this creates a matrix of frames with the original video
            % with a green line around the border
            app.Lb = maskOverlay(rotatedsource,omask,5,[0,0,255]); %
this creates a matrix of frames with the original video
            % with a blue line around the border
            app.viddisplay = zeros(app.R); % this creates a blank
display matrix for use later
            app.viddisplay = app.Ly; % this displays the B&W border
video in the processed video axes
            imshow(app.viddisplay(:,:,1),'Parent',app.UIAxes2);
            [app.cangle, app.cpoints] =
contactAngles(app.B,app.floorheight);
            amask = drawContactAngles(app.R,app.cangle,app.cpoints,
(app.R(2)/4)); % this creates an angle mask for drawing contact angles
matrix
            app.cay = maskOverlay(rotatedsource,amask,5,
[255,255,0]); % this creates a matrix of frames with the original
video
            % with yellow contact angles
            app.car = maskOverlay(rotatedsource,amask,5,[255,0,0]); %
this creates a matrix of frames with the original video
            % with red contact angles
            app.cag = maskOverlay(rotatedsource,amask,5,[0,255,0]); %
this creates a matrix of frames with the original video
            % with green contact angles
            app.cab = maskOverlay(rotatedsource,amask,5,[0,0,255]); %
this creates a matrix of frames with the original video
            % with blue contact angles
            [app.fvelocity, app.impactdata, app.totalobjects] =
fallVelocity(app.B,app.floorheight); %this obtains the fall velocity
of the droplet and jet in the x and y directions
            [app.mspreadl, app.mspreadr, app.spreadl,app.spreadr,
app.ctrmsprdl] = maxSpread(app.B,app.floorheight); % this processes
the B&W border video to find the spread radius per frame and max
spread
            app.MaxRadiusTxt.Value = (app.mspreadl + app.mspreadr)/2;

```

```

        app.TotalSatelliteDropletsTxt.Value =
max(app.totalobjects); % this sets the value of the total satellite
droplets for the video
        [~,app.mxvindex] = max(abs(app.fvelocity(4,1,:)),
[],'omitnan'); % this finds the index of the absolute maximum velocity
        app.MaxVelocityPrimaryDropletTxt.Value =
app.fvelocity(4,1,app.mxvindex); % this sets the value of the
absolute maximum velocity of the primary droplet
        [app.jetv,app.jetpos,app.jetd] = jetVelocity(app.B);
    end

    % Button pushed function: SaveAnalyticsButton
    function SaveAnalyticsButtonPushed(app, event)
        frames = [1:app.R(4)];
        j = 1;
        averageVelocity = nan(1, app.R(4));
        for i = frames
            velocity(i) = app.fvelocity(4,1,i);
            if isnan(app.fvelocity(4,1,i))
            else
                realVelocity(j) = app.fvelocity(4,1,i);
                %All velocity values that are not NaN up to any point i in the loop
                averageVelocity(i) = mean(realVelocity(1:j));
                j = j+1;
            end
            satelliteDropletSpeed(i) = app.fvelocity(4,2,i);
            satelliteDropletCount(i) = app.totalobjects(i);
            contactAngle(i) = mean(app.cangle(:,i));
            spreadRadius(i) = mean([app.spreadl(i)
app.spreadr(i)]);
            jetVelocity(i) = app.jetv(i);
            jetTipPosition(i) = app.jetpos(i);
            jetDiameter(i) = app.jetd(i);
        end

        % Create a table with the data and variable names
        T = table(frames', velocity', averageVelocity',
satelliteDropletSpeed', satelliteDropletCount', contactAngle',
spreadRadius', jetVelocity', jetTipPosition', jetDiameter',...
        'VariableNames',{'Frame Number','Instantaneous
Velocity(um/s)','Average Velocity(um/s)', 'Satellite Droplet
Speed(um/s)', 'Satellite Droplet Count(droplets)', 'Contact
Angle(degrees)', 'Spread Radius(um)', 'Jet Velocity(um/s)', 'Jet Tip
Position(um)', 'Jet Diameter(um)'});
        % Write data to text file
        [txtname,txtpath] = uiputfile('*.csv'); % this asks the
user for a file destination
        fullpath = fullfile(txtpath,txtname); % this concatenates
the path and file name of the file destination
        writetable(T, fullpath);
    end

    % Value changed function: FrameSlider
    function FrameSliderValueChanged(app, event)

```

```

        value = app.FrameSlider.Value;
        imshow(app.M(:, :, :, round(value)), 'Parent', app.UIAxes); %
this displays the selected frame in the original video axes

imshow(app.viddisplay(:, :, :, round(value)), 'Parent', app.UIAxes2); %
this displays the B&W border video in the processed video axes
        app.CurrentFrameEditField.Value = string(round(value)); %
this sets the display in the Current Frame text field
        % to the value selected on the slider
        if isnan(mean(app.cangle(:, round(value)))))
            app.ContactAngleTxt.Visible = 'off';
        else
            app.ContactAngleTxt.Visible = 'on';
            app.ContactAngleTxt.Value =
mean(app.cangle(:, round(value)));
        end

        if isnan(app.fvelocity(4, 1, round(value))) %checking to
see if the velocity at the 'value' frame is a real number
            app.InstantVelocityTxt.Visible = 'off';
        else
            app.InstantVelocityTxt.Visible = 'on';
            app.InstantVelocityTxt.Value =
app.fvelocity(4, 1, round(value));
        end

        if isnan(app.fvelocity(4, 2, round(value))) %checking to
see if the velocity at the 'value' frame is a real number
            app.DropletSpeedTxt.Visible = 'off';
        else
            app.DropletSpeedTxt.Visible = 'on';
            app.DropletSpeedTxt.Value =
app.fvelocity(4, 2, round(value));
        end

        if
isnan(mean(app.fvelocity(4, 1, 1:round(value)), "omitnan")) %checking
to see if the average velocity at the current frame is a real number
            app.AverageVelocityTxt.Visible = 'off';
        else
            app.AverageVelocityTxt.Visible = 'on';
            app.AverageVelocityTxt.Value =
mean(app.fvelocity(4, 1, 1:round(value)), "omitnan");
        end
        if isnan(mean(app.spreadl(round(value)), "omitnan"))
%checking to see if the spread radius at the current frame is a real
number
            app.SpreadRadiusTxt.Visible = 'off';
        else
            app.SpreadRadiusTxt.Visible = 'on';
            app.SpreadRadiusTxt.Value =
mean([app.spreadl(round(value))
app.spreadr(round(value))], "omitnan");
        end

```

```

        if isnan(app.jetv(round(value))) %checking to see if the
jet velocity at the current frame is a real number
            app.JetVelocityTxt.Visible = 'off';
        else
            app.JetVelocityTxt.Visible = 'on';
            app.JetVelocityTxt.Value = app.jetv(round(value));
        end

        if isnan(app.jetpos(round(value))) %checking to see if
the jet tip position at the current frame is a real number
            app.JetTipPositionTxt.Visible = 'off';
        else
            app.JetTipPositionTxt.Visible = 'on';
            app.JetTipPositionTxt.Value =
app.jetpos(round(value));
        end

        if isnan(app.jetd(round(value))) %checking to see if the
jet diameter at the current frame is a real number
            app.JetDiameterTxt.Visible = 'off';
        else
            app.JetDiameterTxt.Visible = 'on';
            app.JetDiameterTxt.Value = app.jetd(round(value));
        end

        app.DropletCountTxt.Value =
app.totalobjects(round(value));
    end

    % Value changing function: FrameSlider
    function FrameSliderValueChanging(app, event)
        changingValue = event.Value;

        imshow(app.M(:, :, :, round(changingValue)), 'Parent', app.UIAxes); % this
displays the selected frame in the original video axes

        imshow(app.viddisplay(:, :, :, round(changingValue)), 'Parent', app.UIAxes2); %
this displays the B&W border video in the processed video axes
        app.CurrentFrameEditField.Value =
string(round(changingValue)); % this sets the display in the Current
Frame text field
        % to the value selected on the slider
        if isnan(mean(app.cangle(:, round(changingValue))))
            app.ContactAngleTxt.Visible = 'off';
        else
            app.ContactAngleTxt.Visible = 'on';
            app.ContactAngleTxt.Value =
mean(app.cangle(:, round(changingValue)));
        end

        if
isnan(app.fvelocity(4,1,round(changingValue))) %checking to see if
the velocity at the 'changingValue' frame is a real number

```

```

        app.InstantVelocityTxt.Visible = 'off';
    else
        app.InstantVelocityTxt.Visible = 'on';
        app.InstantVelocityTxt.Value =
app.fvelocity(4,1,round(changingValue));
    end

    if
isnan(app.fvelocity(4,2,round(changingValue))) %checking to see if
the velocity at the 'changingValue' frame is a real number
        app.DropletSpeedTxt.Visible = 'off';
    else
        app.DropletSpeedTxt.Visible = 'on';
        app.DropletSpeedTxt.Value =
app.fvelocity(4,2,round(changingValue));
    end

    if
isnan(mean(app.fvelocity(4,1,1:round(changingValue)),"omitnan"))
    %checking to see if the average velocity at the current frame is a
real number
        app.AverageVelocityTxt.Visible = 'off';
    else
        app.AverageVelocityTxt.Visible = 'on';
        app.AverageVelocityTxt.Value =
mean(app.fvelocity(4,1,1:round(changingValue)),"omitnan");
    end

    if
isnan(mean(app.spreadl(round(changingValue)),"omitnan")) %checking
to see if the spread radius at the current frame is a real number
        app.SpreadRadiusTxt.Visible = 'off';
    else
        app.SpreadRadiusTxt.Visible = 'on';
        app.SpreadRadiusTxt.Value
= mean([app.spreadl(round(changingValue))
app.spreadr(round(changingValue))],"omitnan");
    end

    if isnan(app.jetv(round(changingValue))) %checking to
see if the jet velocity at the current frame is a real number
        app.JetVelocityTxt.Visible = 'off';
    else
        app.JetVelocityTxt.Visible = 'on';
        app.JetVelocityTxt.Value =
app.jetv(round(changingValue));
    end

    if isnan(app.jetpos(round(changingValue))) %checking to
see if the jet tip position at the current frame is a real number
        app.JetTipPositionTxt.Visible = 'off';
    else
        app.JetTipPositionTxt.Visible = 'on';

```

```

        app.JetTipPositionTxt.Value =
app.jetpos(round(changingValue));
    end

    if isnan(app.jetd(round(changingValue))) %checking to
see if the jet diameter at the current frame is a real number
        app.JetDiameterTxt.Visible = 'off';
    else
        app.JetDiameterTxt.Visible = 'on';
        app.JetDiameterTxt.Value =
app.jetd(round(changingValue));
    end

    app.DropletCountTxt.Value =
app.totalobjects(round(changingValue));
    end

% Button pushed function: SaveProcessedVideoButton
function SaveProcessedVideoButtonPushed(app, event)
    [filename,filepath]=uinputfile('.JPEG');
    [~,~,~,sizeapp] = size(app.L);
    i=1;
    while(i <= sizeapp)
        frame2file(app.L,filename,filepath,i);
        i=i+1;
    end
end

% Value changed function: MicronspixelEditField
function MicronspixelEditFieldValueChanged(app, event)
    app.micron_to_pxl = app.MicronspixelEditField.Value; %
collects user input for pxl/mm
    if app.micron_to_pxl == 0
        f = errordlg('Pixels/mm Cannot Be Set to 0. Setting
Pixels/mm to 1...', 'Invalid Pixels/Millimeter Input');
        app.MicronspixelEditField.Value = 1;
    end
    app.mspreadl = app.mspreadl*app.micron_to_pxl; % converts
max spread radius left
    app.mspreadr = app.mspreadr*app.micron_to_pxl; % converts
max spread radius right
    app.spreadl = app.spreadl*app.micron_to_pxl; % converts
spread radius left
    app.spreadr = app.spreadr*app.micron_to_pxl; % converts
spread radius right
    app.fvelocity([3 4],:,:) = app.fvelocity([3
4],:,:)*app.micron_to_pxl; % converts fall velocity of droplet x and
y directions
    app.jetv = app.jetv*app.micron_to_pxl; % converts jet
velocity
    app.jetpos = app.jetpos*app.micron_to_pxl; % converts
height of the tip of the jet
    app.jetd = app.jetd*app.micron_to_pxl; % converts jet
diameter

```

```

        value = app.FrameSlider.Value; % finds current frame for
        updating data currently displayed
        app.MaxRadiusTxt.Value = (app.mspreadl +
        app.mspreadr)/2; % updates the display of maxspread
        app.MaxVelocityPrimaryDropletTxt.Value =
        app.fvelocity(4,1,app.mxvindex); % updates the absolute maximum
        velocity of the primary droplet

        if isnan(app.fvelocity(4,1,round(value))) %checking to
        see if the velocity at the 'value' frame is a real number
            app.InstantVelocityTxt.Visible = 'off';
        else
            app.InstantVelocityTxt.Visible = 'on';
            app.InstantVelocityTxt.Value =
            app.fvelocity(4,1,round(value));
        end

        if isnan(app.fvelocity(4,2,round(value))) %checking to
        see if the velocity at the 'value' frame is a real number
            app.DropletSpeedTxt.Visible = 'off';
        else
            app.DropletSpeedTxt.Visible = 'on';
            app.DropletSpeedTxt.Value =
            app.fvelocity(4,2,round(value));
        end

        if
        isnan(mean(app.fvelocity(4,1,1:round(value)),"omitnan")) %checking
        to see if the average velocity at the current frame is a real number
            app.AverageVelocityTxt.Visible = 'off';
        else
            app.AverageVelocityTxt.Visible = 'on';
            app.AverageVelocityTxt.Value =
            mean(app.fvelocity(4,1,1:round(value)),"omitnan");
        end
        if isnan(mean(app.spreadl(round(value)),"omitnan"))
        %checking to see if the spread radius at the current frame is a real
        number
            app.SpreadRadiusTxt.Visible = 'off';
        else
            app.SpreadRadiusTxt.Visible = 'on';
            app.SpreadRadiusTxt.Value =
            mean([app.spreadl(round(value))
            app.spreadr(round(value))],"omitnan");
        end

        if isnan(app.jetv(round(value))) %checking to see if the
        jet velocity at the current frame is a real number
            app.JetVelocityTxt.Visible = 'off';
        else
            app.JetVelocityTxt.Visible = 'on';
            app.JetVelocityTxt.Value = app.jetv(round(value));
        end

```

```

        if isnan(app.jetpos(round(value)))    %checking to see if
the jet tip position at the current frame is a real number
        app.JetTipPositionTxt.Visible = 'off';
    else
        app.JetTipPositionTxt.Visible = 'on';
        app.JetTipPositionTxt.Value =
app.jetpos(round(value));
    end

    if isnan(app.jetd(round(value)))    %checking to see if the
jet diameter at the current frame is a real number
        app.JetDiameterTxt.Visible = 'off';
    else
        app.JetDiameterTxt.Visible = 'on';
        app.JetDiameterTxt.Value = app.jetd(round(value));
    end
end

% Value changed function: OutlineColorDropDown
function OutlineColorDropDownValueChanged(app, event)
    currentframe = round(app.FrameSlider.Value);
    value = app.OutlineColorDropDown.Value;
    cavalue = app.ContactAnglesOffButton.Value;
    if cavalue == 0
        if strcmp(value, 'Red')
            app.viddisplay = app.Lr; % this sets the displayed
video as the red outline video
        elseif strcmp(value, 'Blue')
            app.viddisplay = app.Lb; % this sets the displayed
video as the blue outline video
        elseif strcmp(value, 'Yellow')
            app.viddisplay = app.Ly; % this sets the displayed
video as the yellow outline video
        elseif strcmp(value, 'Green')
            app.viddisplay = app.Lg; % this sets the displayed
video as the green outline video
        end
    elseif cavalue == 1
        if strcmp(value, 'Red')
            app.viddisplay = app.car; % this sets the
displayed video as the red angle video
        elseif strcmp(value, 'Blue')
            app.viddisplay = app.cab; % this sets the
displayed video as the blue angle video
        elseif strcmp(value, 'Yellow')
            app.viddisplay = app.cay; % this sets the
displayed video as the yellow angle video
        elseif strcmp(value, 'Green')
            app.viddisplay = app.cag; % this sets the
displayed video as the green angle video
        end
    end
end

```

```

imshow(app.viddisplay(:,:,currentframe),'Parent',app.UIAxes2); %
this displays the B&W border video in the processed video axes
end

% Button pushed function: SaveCurrentFrameButton
function SaveCurrentFrameButtonPushed(app, event)
    currentframe = round(app.FrameSlider.Value);
    [filename,filepath]=uiputfile('.JPEG');

frame2file(app.viddisplay,filename,filepath,currentframe); % this
saves the current frame of the processed video axes
end

% Value changed function: ContactAnglesOffButton
function ContactAnglesOffButtonValueChanged(app, event)
    currentframe = round(app.FrameSlider.Value);
    value = app.ContactAnglesOffButton.Value;
    colvalue = app.OutlineColorDropDown.Value;
    if value == 0
        app.ContactAnglesOffButton.Text = 'Contact Angles
Off';
        if strcmp(colvalue,'Red')
            app.viddisplay = app.Lr; % this sets the displayed
video as the red outline video
        elseif strcmp(colvalue,'Blue')
            app.viddisplay = app.Lb; % this sets the displayed
video as the blue outline video
        elseif strcmp(colvalue,'Yellow')
            app.viddisplay = app.Ly; % this sets the displayed
video as the yellow outline video
        elseif strcmp(colvalue,'Green')
            app.viddisplay = app.Lg; % this sets the displayed
video as the green outline video
        end
    elseif value == 1
        app.ContactAnglesOffButton.Text = 'Contact Angles On';
        if strcmp(colvalue,'Red')
            app.viddisplay = app.car; % this sets the
displayed video as the red angle video
        elseif strcmp(colvalue,'Blue')
            app.viddisplay = app.cab; % this sets the
displayed video as the blue angle video
        elseif strcmp(colvalue,'Yellow')
            app.viddisplay = app.cay; % this sets the
displayed video as the yellow angle video
        elseif strcmp(colvalue,'Green')
            app.viddisplay = app.cag; % this sets the
displayed video as the green angle video
        end
    end
end

imshow(app.viddisplay(:,:,currentframe),'Parent',app.UIAxes2); %
this displays the B&W border video in the processed video axes

```

```

end

% Value changed function: FramesecEditField
function FramesecEditFieldValueChanged(app, event)
    app.frame_to_s = app.FramesecEditField.Value;
    if app.frame_to_s == 0
        f = errordlg('Frame/Sec Cannot Be Set to 0. Setting
Frame/s to 1...', 'Invalid Frame/Second Input');
        app.frame_to_s = 1;
    end
    app.fvelocity([3 4], :, :) = app.fvelocity([3
4], :, :)*app.frame_to_s; % converts fall velocity of droplet x and y
directions
    app.jetv = app.jetv*app.frame_to_s; % converts jet
velocity

    value = app.FrameSlider.Value; % finds current frame for
updating data currently displayed
    app.MaxVelocityPrimaryDropletTxt.Value =
app.fvelocity(4,1,app.mxvindex); % updates the absolute maximum
velocity of the primary droplet

    if isnan(app.fvelocity(4,1,round(value))) %checking to
see if the velocity at the 'value' frame is a real number
        app.InstantVelocityTxt.Visible = 'off';
    else
        app.InstantVelocityTxt.Visible = 'on';
        app.InstantVelocityTxt.Value =
app.fvelocity(4,1,round(value));
    end

    if isnan(app.fvelocity(4,2,round(value))) %checking to
see if the velocity at the 'value' frame is a real number
        app.DropletSpeedTxt.Visible = 'off';
    else
        app.DropletSpeedTxt.Visible = 'on';
        app.DropletSpeedTxt.Value =
app.fvelocity(4,2,round(value));
    end

    if
isnan(mean(app.fvelocity(4,1,1:round(value)), 'omitnan')) %checking
to see if the average velocity at the current frame is a real number
        app.AverageVelocityTxt.Visible = 'off';
    else
        app.AverageVelocityTxt.Visible = 'on';
        app.AverageVelocityTxt.Value =
mean(app.fvelocity(4,1,1:round(value)), 'omitnan');
    end
    if isnan(app.jetv(round(value))) %checking to see if the
jet velocity at the current frame is a real number
        app.JetVelocityTxt.Visible = 'off';
    else
        app.JetVelocityTxt.Visible = 'on';

```

```

        app.JetVelocityTxt.Value = app.jetv(round(value));
    end
end
end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create UIFigure and hide until all components are
created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 653 718];
        app.UIFigure.Name = 'MATLAB App';

        % Create UploadVideoButton
        app.UploadVideoButton = uibutton(app.UIFigure, 'push');
        app.UploadVideoButton.ButtonPushedFcn =
createCallbackFcn(app, @UploadVideoButtonPushed, true);
        app.UploadVideoButton.Position = [124 496 100 22];
        app.UploadVideoButton.Text = 'Upload Video';

        % Create FrameSliderLabel
        app.FrameSliderLabel = uilabel(app.UIFigure);
        app.FrameSliderLabel.HorizontalAlignment = 'right';
        app.FrameSliderLabel.Position = [313 340 40 22];
        app.FrameSliderLabel.Text = 'Frame';

        % Create FrameSlider
        app.FrameSlider = uislider(app.UIFigure);
        app.FrameSlider.Limits = [0 15];
        app.FrameSlider.MajorTicks = [5 10 15];
        app.FrameSlider.ValueChangedFcn = createCallbackFcn(app,
@FrameSliderValueChanged, true);
        app.FrameSlider.ValueChangingFcn = createCallbackFcn(app,
@FrameSliderValueChanging, true);
        app.FrameSlider.MinorTicks = [1 2 3 4 5 6 7 8 9 10 11 12
13 14 15];
        app.FrameSlider.Position = [88 384 479 3];

        % Create SaveProcessedVideoButton
        app.SaveProcessedVideoButton =
uibutton(app.UIFigure, 'push');
        app.SaveProcessedVideoButton.ButtonPushedFcn =
createCallbackFcn(app, @SaveProcessedVideoButtonPushed, true);
        app.SaveProcessedVideoButton.Position = [466 38 140 22];
        app.SaveProcessedVideoButton.Text = 'Save Processed Video
';

        % Create InstantaneousVelocityButton
        app.InstantaneousVelocityButton =
uibutton(app.UIFigure, 'push');

```

```

app.InstantaneousVelocityButton.Position = [70 296 131
22];
app.InstantaneousVelocityButton.Text = 'Instantaneous
Velocity';

% Create AverageVelocityButton
app.AverageVelocityButton =
uibutton(app.UIFigure, 'push');
app.AverageVelocityButton.Position = [83 261 105 22];
app.AverageVelocityButton.Text = 'Average Velocity';

% Create MaxSpreadRadiusButton
app.MaxSpreadRadiusButton =
uibutton(app.UIFigure, 'push');
app.MaxSpreadRadiusButton.Position = [466 163 121 22];
app.MaxSpreadRadiusButton.Text = 'Max Spread Radius';

% Create SaveAnalyticsButton
app.SaveAnalyticsButton = uibutton(app.UIFigure, 'push');
app.SaveAnalyticsButton.ButtonPushedFcn =
createCallbackFcn(app, @SaveAnalyticsButtonPushed, true);
app.SaveAnalyticsButton.Position = [486 75 100 22];
app.SaveAnalyticsButton.Text = 'Save Analytics';

% Create SatelliteDropletSpeedButton
app.SatelliteDropletSpeedButton =
uibutton(app.UIFigure, 'push');
app.SatelliteDropletSpeedButton.Position = [71 228 132
22];
app.SatelliteDropletSpeedButton.Text = 'Satellite Droplet
Speed';

% Create SatelliteDropletCountButton
app.SatelliteDropletCountButton =
uibutton(app.UIFigure, 'push');
app.SatelliteDropletCountButton.Position = [71 196 129
22];
app.SatelliteDropletCountButton.Text = 'Satellite Droplet
Count';

% Create InstantaneousVelocityUnits
app.InstantaneousVelocityUnits =
uieditfield(app.UIFigure, 'text');
app.InstantaneousVelocityUnits.Editable = 'off';
app.InstantaneousVelocityUnits.BackgroundColor = [0.9412
0.9412 0.9412];
app.InstantaneousVelocityUnits.Position = [300 294 100
22];
app.InstantaneousVelocityUnits.Value = 'um/s';

% Create AverageVelocityUnits
app.AverageVelocityUnits =
uieditfield(app.UIFigure, 'text');
app.AverageVelocityUnits.Editable = 'off';

```

```

app.AverageVelocityUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.AverageVelocityUnits.Position = [300 261 100 22];
app.AverageVelocityUnits.Value = 'um/s';

% Create MaxSpreadUnits
app.MaxSpreadUnits = uieditfield(app.UIFigure, 'text');
app.MaxSpreadUnits.Editable = 'off';
app.MaxSpreadUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.MaxSpreadUnits.Position = [561 128 62 22];
app.MaxSpreadUnits.Value = 'um';

% Create DropletSpeedUnits
app.DropletSpeedUnits = uieditfield(app.UIFigure, 'text');
app.DropletSpeedUnits.Editable = 'off';
app.DropletSpeedUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.DropletSpeedUnits.Position = [300 228 100 22];
app.DropletSpeedUnits.Value = 'um/s';

% Create DropletCountUnits
app.DropletCountUnits = uieditfield(app.UIFigure, 'text');
app.DropletCountUnits.Editable = 'off';
app.DropletCountUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.DropletCountUnits.Position = [300 196 100 22];
app.DropletCountUnits.Value = 'droplets';

% Create CurrentFrameEditFieldLabel
app.CurrentFrameEditFieldLabel = uilabel(app.UIFigure);
app.CurrentFrameEditFieldLabel.HorizontalAlignment
= 'right';
app.CurrentFrameEditFieldLabel.Position = [86 412 83 22];
app.CurrentFrameEditFieldLabel.Text = 'Current Frame';

% Create CurrentFrameEditField
app.CurrentFrameEditField =
uieditfield(app.UIFigure, 'text');
app.CurrentFrameEditField.Editable = 'off';
app.CurrentFrameEditField.Position = [184 412 62 17];
app.CurrentFrameEditField.Value = '#';

% Create InstantVelocityTxt
app.InstantVelocityTxt =
uieditfield(app.UIFigure, 'numeric');
app.InstantVelocityTxt.Position = [231 294 44 27];

% Create AverageVelocityTxt
app.AverageVelocityTxt =
uieditfield(app.UIFigure, 'numeric');
app.AverageVelocityTxt.Position = [231 259 44 27];

% Create MaxRadiusTxt

```

```

        app.MaxRadiusTxt = uieditfield(app.UIFigure, 'numeric');
        app.MaxRadiusTxt.Position = [504 126 44 27];

        % Create DropletSpeedTxt
        app.DropletSpeedTxt =
uieditfield(app.UIFigure, 'numeric');
        app.DropletSpeedTxt.Position = [231 226 44 27];

        % Create DropletCountTxt
        app.DropletCountTxt =
uieditfield(app.UIFigure, 'numeric');
        app.DropletCountTxt.Position = [231 194 44 27];

        % Create MicronspixelEditFieldLabel
        app.MicronspixelEditFieldLabel = uilabel(app.UIFigure);
        app.MicronspixelEditFieldLabel.HorizontalAlignment
= 'right';
        app.MicronspixelEditFieldLabel.Position = [25 454 75 22];
        app.MicronspixelEditFieldLabel.Text = 'Microns/pixel';

        % Create MicronspixelEditField
        app.MicronspixelEditField =
uieditfield(app.UIFigure, 'numeric');
        app.MicronspixelEditField.ValueChangedFcn =
createCallbackFcn(app, @MicronspixelEditFieldValueChanged, true);
        app.MicronspixelEditField.Position = [109 454 64 22];
        app.MicronspixelEditField.Value = 1;

        % Create ContactAngleButton
        app.ContactAngleButton = uibutton(app.UIFigure, 'push');
        app.ContactAngleButton.Position = [85 161 100 22];
        app.ContactAngleButton.Text = 'Contact Angle';

        % Create ContactAngleUnits
        app.ContactAngleUnits = uieditfield(app.UIFigure, 'text');
        app.ContactAngleUnits.Editable = 'off';
        app.ContactAngleUnits.BackgroundColor = [0.9412 0.9412
0.9412];
        app.ContactAngleUnits.Position = [300 161 100 22];
        app.ContactAngleUnits.Value = 'degrees';

        % Create ContactAngleTxt
        app.ContactAngleTxt =
uieditfield(app.UIFigure, 'numeric');
        app.ContactAngleTxt.Position = [231 159 44 27];

        % Create TotalSatelliteDropletsButton
        app.TotalSatelliteDropletsButton =
uibutton(app.UIFigure, 'push');
        app.TotalSatelliteDropletsButton.Position = [458 293 135
28];
        app.TotalSatelliteDropletsButton.Text = 'Total Satellite
Droplets';

```

```

        % Create TotalSatelliteDropletsTxt
        app.TotalSatelliteDropletsTxt =
uieditfield(app.UIFigure, 'numeric');
        app.TotalSatelliteDropletsTxt.Position = [504 259 44 27];

        % Create MaxVelocityPrimaryDropletButton
        app.MaxVelocityPrimaryDropletButton =
uibutton(app.UIFigure, 'push');
        app.MaxVelocityPrimaryDropletButton.Position = [439 228
178 22];
        app.MaxVelocityPrimaryDropletButton.Text = 'Max Velocity
(Max Velocity Primary Droplet)';

        % Create MaxVelocityPrimaryDropletTxt
        app.MaxVelocityPrimaryDropletTxt =
uieditfield(app.UIFigure, 'numeric');
        app.MaxVelocityPrimaryDropletTxt.Position = [504 193 44
27];

        % Create MaxVelocityPrimaryDropletUnits
        app.MaxVelocityPrimaryDropletUnits =
uieditfield(app.UIFigure, 'text');
        app.MaxVelocityPrimaryDropletUnits.Editable = 'off';
        app.MaxVelocityPrimaryDropletUnits.BackgroundColor =
[0.9412 0.9412 0.9412];
        app.MaxVelocityPrimaryDropletUnits.Position = [561 195 62
22];
        app.MaxVelocityPrimaryDropletUnits.Value = 'um/s';

        % Create TotalSatelliteDropletsUnits
        app.TotalSatelliteDropletsUnits =
uieditfield(app.UIFigure, 'text');
        app.TotalSatelliteDropletsUnits.Editable = 'off';
        app.TotalSatelliteDropletsUnits.BackgroundColor = [0.9412
0.9412 0.9412];
        app.TotalSatelliteDropletsUnits.Position = [561 261 62
22];
        app.TotalSatelliteDropletsUnits.Value = 'droplets';

        % Create OutlineColorDropDownLabel
        app.OutlineColorDropDownLabel = uilabel(app.UIFigure);
        app.OutlineColorDropDownLabel.HorizontalAlignment
= 'right';
        app.OutlineColorDropDownLabel.Position = [478 433 75 22];
        app.OutlineColorDropDownLabel.Text = 'Outline Color';

        % Create OutlineColorDropDown
        app.OutlineColorDropDown = uidropdown(app.UIFigure);
        app.OutlineColorDropDown.Items =
{'Yellow', 'Green', 'Blue', 'Red'};
        app.OutlineColorDropDown.ValueChangedFcn =
createCallbackFcn(app, @OutlineColorDropDownValueChanged, true);
        app.OutlineColorDropDown.Position = [466 412 100 22];
        app.OutlineColorDropDown.Value = 'Yellow';

```

```

        % Create SaveCurrentFrameButton
        app.SaveCurrentFrameButton =
uibutton(app.UIFigure, 'push');
        app.SaveCurrentFrameButton.ButtonPushedFcn =
createCallbackFcn(app, @SaveCurrentFrameButtonPushed, true);
        app.SaveCurrentFrameButton.Position = [451 496 124 22];
        app.SaveCurrentFrameButton.Text = 'Save Current Frame';

        % Create SpreadRadius
        app.SpreadRadius = uibutton(app.UIFigure, 'push');
        app.SpreadRadius.Position = [86 124 100 22];
        app.SpreadRadius.Text = 'Spread Radius';

        % Create SpreadRadiusUnits
        app.SpreadRadiusUnits = uieditfield(app.UIFigure, 'text');
        app.SpreadRadiusUnits.Editable = 'off';
        app.SpreadRadiusUnits.BackgroundColor = [0.9412 0.9412
0.9412];
        app.SpreadRadiusUnits.Position = [300 124 100 22];
        app.SpreadRadiusUnits.Value = 'um';

        % Create SpreadRadiusTxt
        app.SpreadRadiusTxt =
uieditfield(app.UIFigure, 'numeric');
        app.SpreadRadiusTxt.Position = [231 122 44 27];

        % Create ContactAnglesOffButton
        app.ContactAnglesOffButton =
uibutton(app.UIFigure, 'state');
        app.ContactAnglesOffButton.ValueChangedFcn =
createCallbackFcn(app, @ContactAnglesOffButtonValueChanged, true);
        app.ContactAnglesOffButton.Text = 'Contact Angles Off';
        app.ContactAnglesOffButton.Position = [455 463 116 22];

        % Create JetVelocity
        app.JetVelocity = uibutton(app.UIFigure, 'push');
        app.JetVelocity.Position = [101 89 71 22];
        app.JetVelocity.Text = 'Jet Velocity';

        % Create JetVelocityUnits
        app.JetVelocityUnits = uieditfield(app.UIFigure, 'text');
        app.JetVelocityUnits.Editable = 'off';
        app.JetVelocityUnits.BackgroundColor = [0.9412 0.9412
0.9412];
        app.JetVelocityUnits.Position = [300 89 100 22];
        app.JetVelocityUnits.Value = 'um/s';

        % Create JetVelocityTxt
        app.JetVelocityTxt = uieditfield(app.UIFigure, 'numeric');
        app.JetVelocityTxt.Position = [231 87 44 27];

        % Create JetTipPosition
        app.JetTipPosition = uibutton(app.UIFigure, 'push');

```

```

app.JetTipPosition.Position = [85 54 100 22];
app.JetTipPosition.Text = 'Jet Tip Position';

% Create JetTipPositionUnits
app.JetTipPositionUnits =
uieditfield(app.UIFigure, 'text');
app.JetTipPositionUnits.Editable = 'off';
app.JetTipPositionUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.JetTipPositionUnits.Position = [300 54 100 22];
app.JetTipPositionUnits.Value = 'um';

% Create JetTipPositionTxt
app.JetTipPositionTxt =
uieditfield(app.UIFigure, 'numeric');
app.JetTipPositionTxt.Position = [231 52 44 27];

% Create JetDiameter
app.JetDiameter = uibutton(app.UIFigure, 'push');
app.JetDiameter.Position = [86 17 100 22];
app.JetDiameter.Text = 'Jet Diameter';

% Create JetDiameterUnits
app.JetDiameterUnits = uieditfield(app.UIFigure, 'text');
app.JetDiameterUnits.Editable = 'off';
app.JetDiameterUnits.BackgroundColor = [0.9412 0.9412
0.9412];
app.JetDiameterUnits.Position = [300 17 100 22];
app.JetDiameterUnits.Value = 'um';

% Create JetDiameterTxt
app.JetDiameterTxt = uieditfield(app.UIFigure, 'numeric');
app.JetDiameterTxt.Position = [231 15 44 27];

% Create FramessecLabel
app.FramessecLabel = uilabel(app.UIFigure);
app.FramessecLabel.HorizontalAlignment = 'right';
app.FramessecLabel.Position = [181 454 68 22];
app.FramessecLabel.Text = 'Frames/sec';

% Create FramessecEditField
app.FramessecEditField =
uieditfield(app.UIFigure, 'numeric');
app.FramessecEditField.ValueChangedFcn =
createCallbackFcn(app, @FramessecEditFieldValueChanged, true);
app.FramessecEditField.Position = [258 454 65 22];
app.FramessecEditField.Value = 1;

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, 'Original Video')
app.UIAxes.PlotBoxAspectRatio = [1.93129770992366 1 1];
app.UIAxes.Position = [1 517 300 185];

```

```
% Create UIAxes2
app.UIAxes2 = uiaxes(app.UIFigure);
title(app.UIAxes2, 'Processed Video')
app.UIAxes2.Position = [334 517 300 185];

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = GUI_Design_exported

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)
end
end
end
```

Published with MATLAB® R2020b