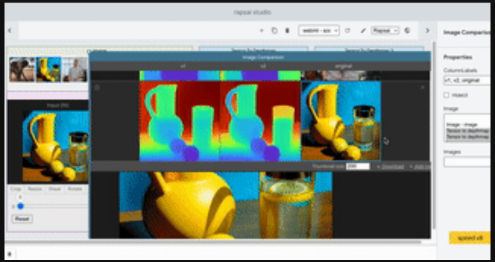
Honorable Mention

rapsai

Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming



Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan,
Yinda Zhang, Anuva Kulkarni, Xingyu "Bruce" Liu, Ahmed Sabie, Sergio Escolano, Abhishek Kar,
Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal
visualblocksforml.github.io

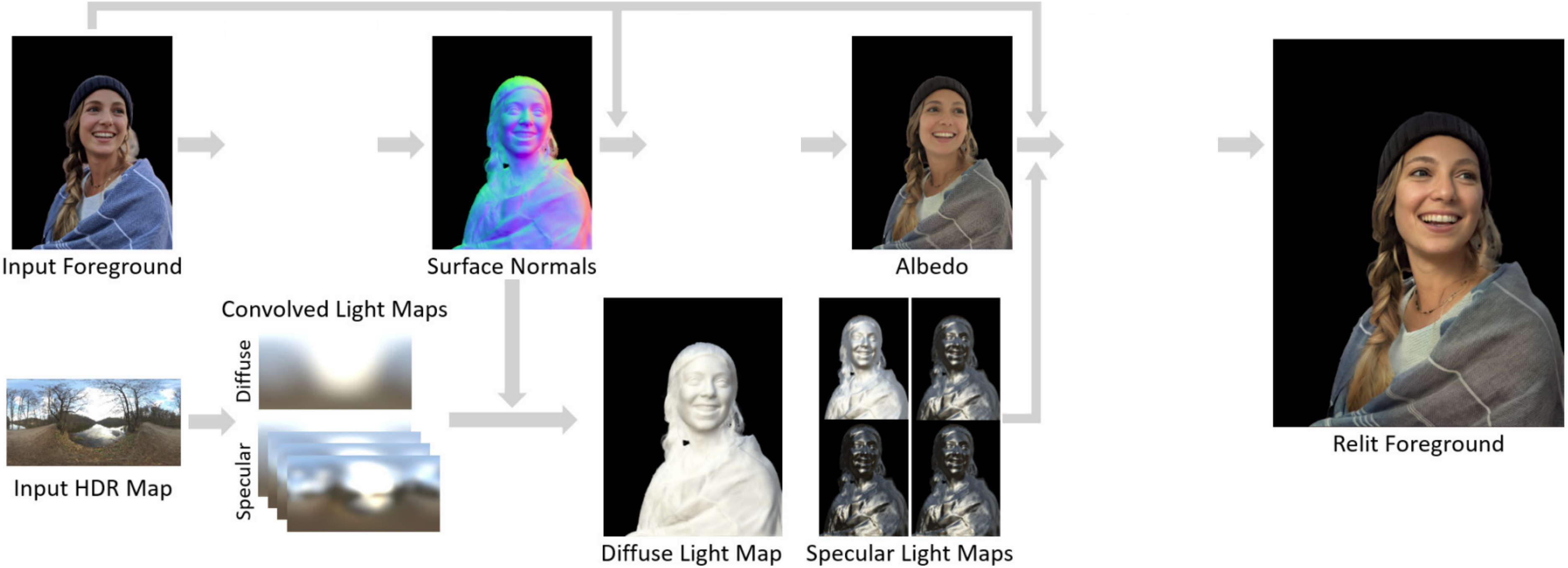
Google

CHI23
Hamburg, Germany | Hybrid
April 23-28, 2023
reCHInnecting

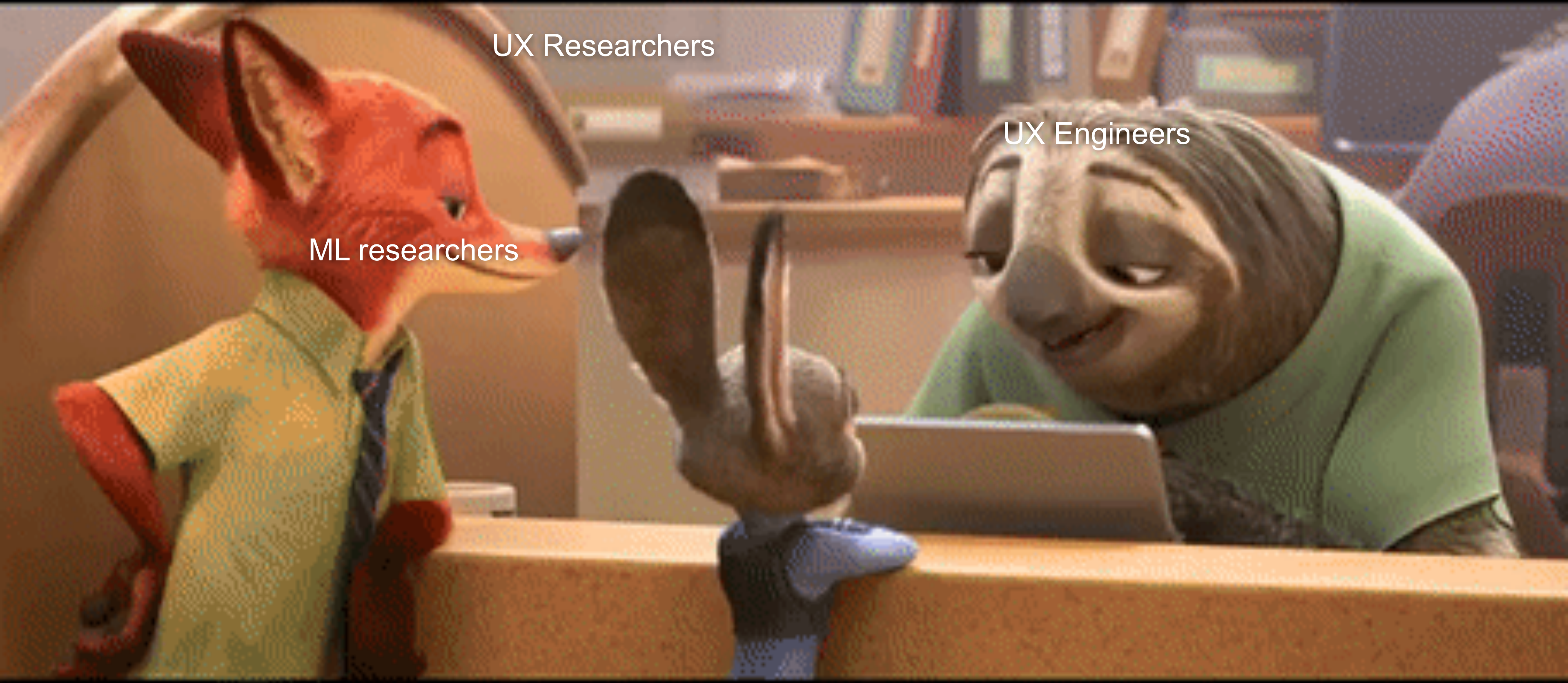

3D Photos


Portrait Relighting


Virtual Background
MediaPipe



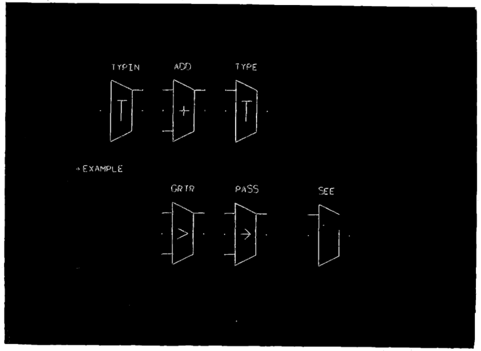
Pandey, Rohit, Sergio Orts Escolano, Chloe Legendre, Christian Haene, Sofien Bouaziz, Christoph Rhemann, Paul Debevec, and Sean Fanello. "Total relighting: learning to relight portraits for background replacement." ACM Transactions on Graphics (SIGGRAPH 2021) 40, no. 4 (2021): 1-21.


UX Researchers
UX Engineers
ML researchers

Related Work

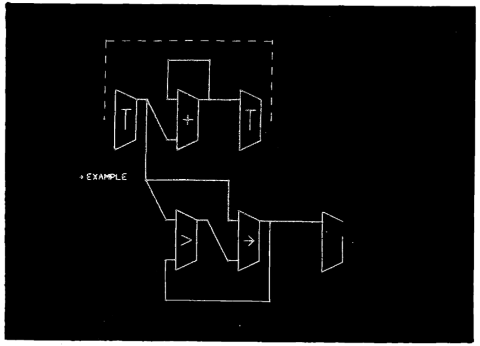
Visual programming in language

Related Work
Visual programming in graphics



Basic Symbols

Figure 1.1



Connected Program

Figure 1.2

12

THE ON-LINE GRAPHICAL SPECIFICATION OF COMPUTER PROCEDURES

by

WILLIAM ROBERT SUTHERLAND

B.E.E., Rensselaer Polytechnic Institute

(1957)

M.S., Massachusetts Institute of Technology

(1963)

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January, 1966

Signature of Author..
Department of Electrical Engineering, January 10, 1966

Certified by..
Thesis Supervisor

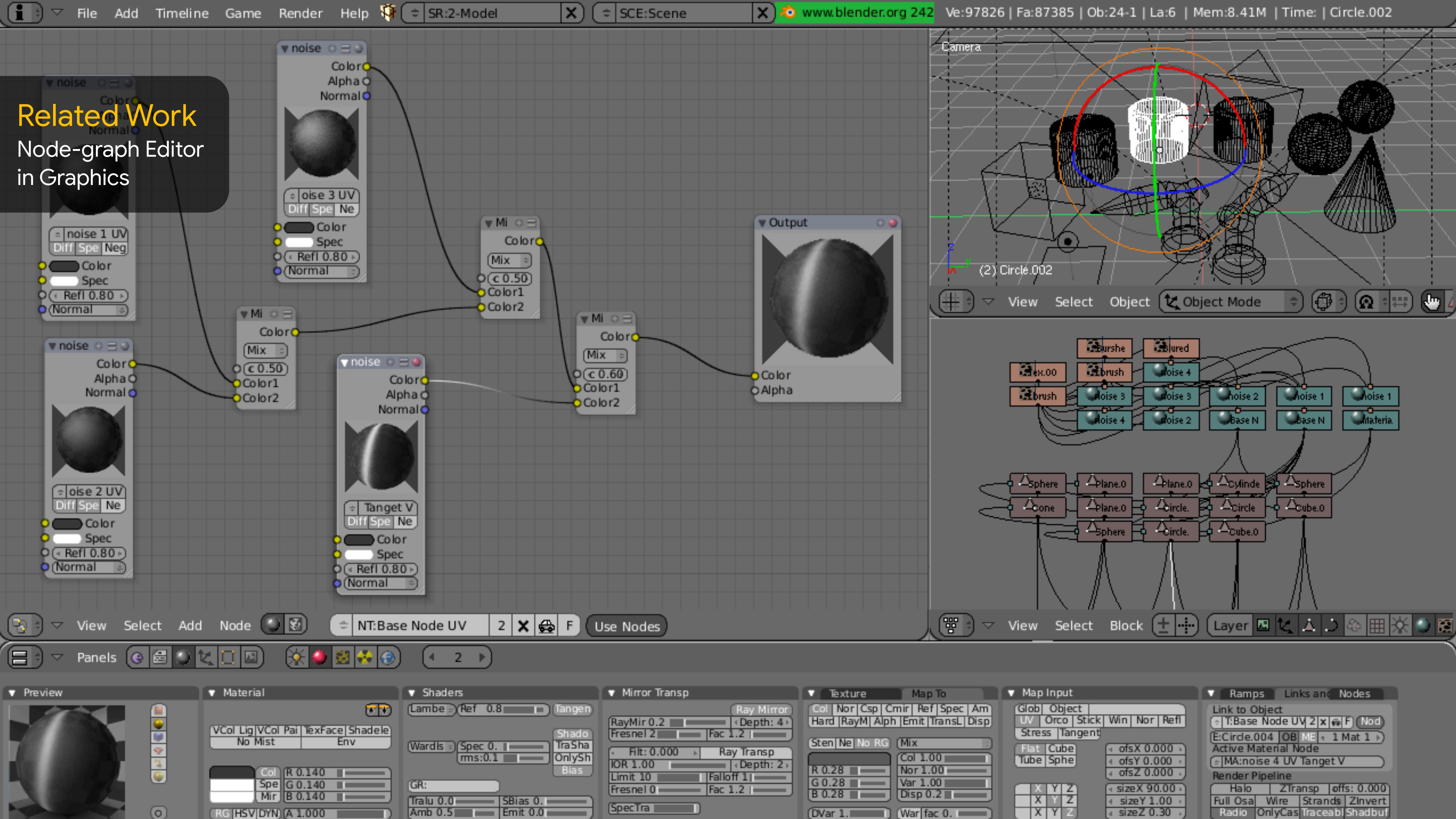
Accepted by...
Chairman, Departmental Committee on Graduate Students

1

WR Sutherland. 1966. On-Line Graphical Specification of Procedures. SJCC, Boston, Mass (1966).

Related Work

Node-graph Editor in Graphics




3D Photos


Portrait Relighting


Virtual Background
MediaPipe

person 0.878
baseball glove 0.157
person 0.888
person 0.864
baseball glove 0.769
Id: 2
potted plant,0.62
Id: 1
laptop,0.86

TensorFlow

PyTorch

Keras

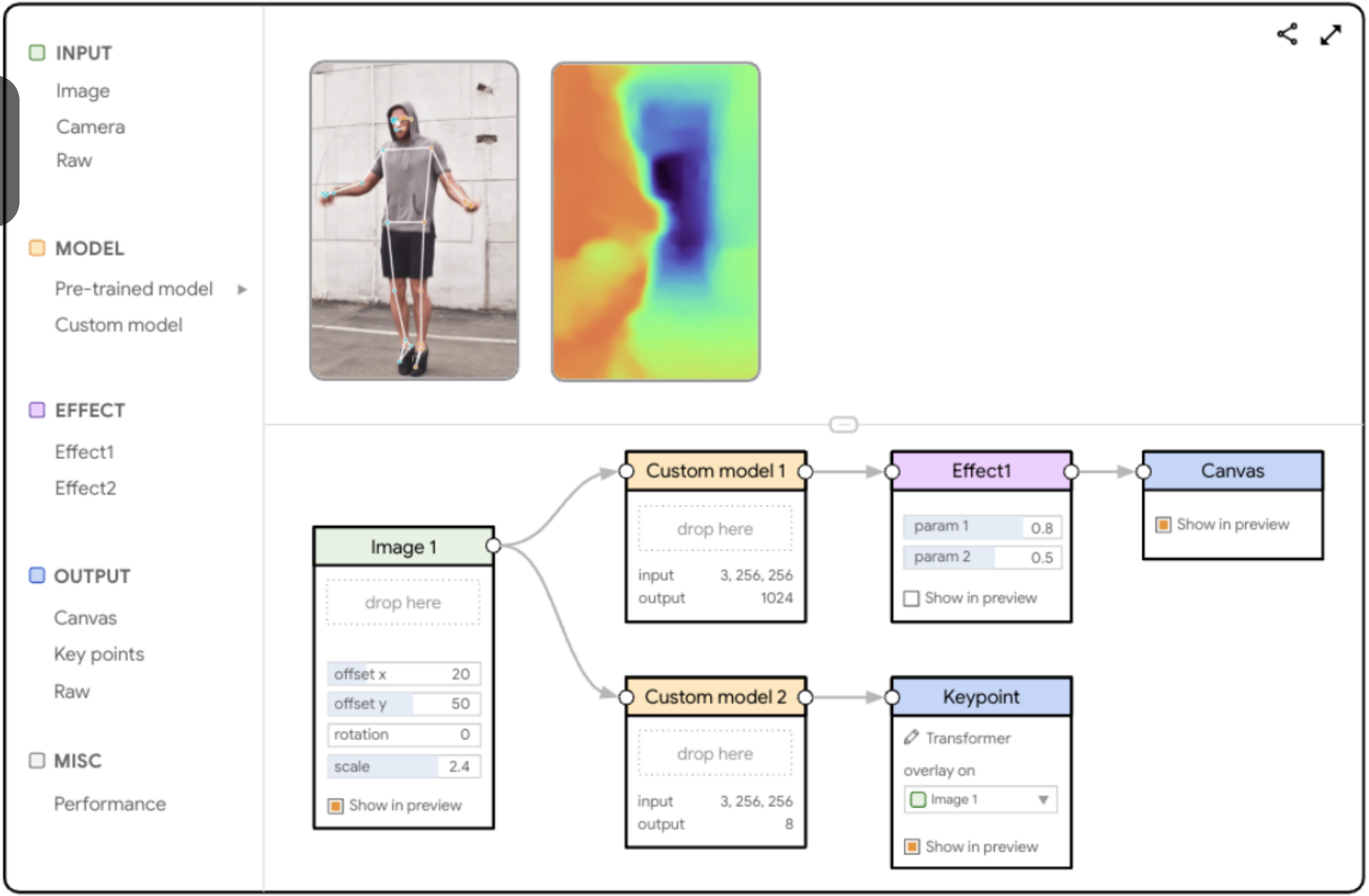
OpenGL

Vulkan

Model Development → **Video / Graphics / Audio Processing** → **Multimedia Applications**


Formative Study (N=7)
Jing Jin
Na Li

Mockup Sketch



Lack of data processing for input in-the-wild

1

The complexity of processing of input formats yields longer cycles between finding failure cases and fine-tuning new models.

Lack of interactive data and model tuning

2

"models can be affected by over-fitting, so need to test with a large variety of image augmentations" (I2)

Loss of application context

3

“Metric doesn’t help <in my depth models>, it’s always good for all the models, so it’s no use. They need human eyes to evaluate.” (I6)

Lack of direct comparison and sharing

4

"I want to isolate bad examples of a specific error pattern to discuss with stakeholders." (I1)

Slow iterations

5

"A lot of time goes into visualization of challenge sets, benchmarking, and metrics. Usually takes weeks." (I2)

Insufficient controllability

6

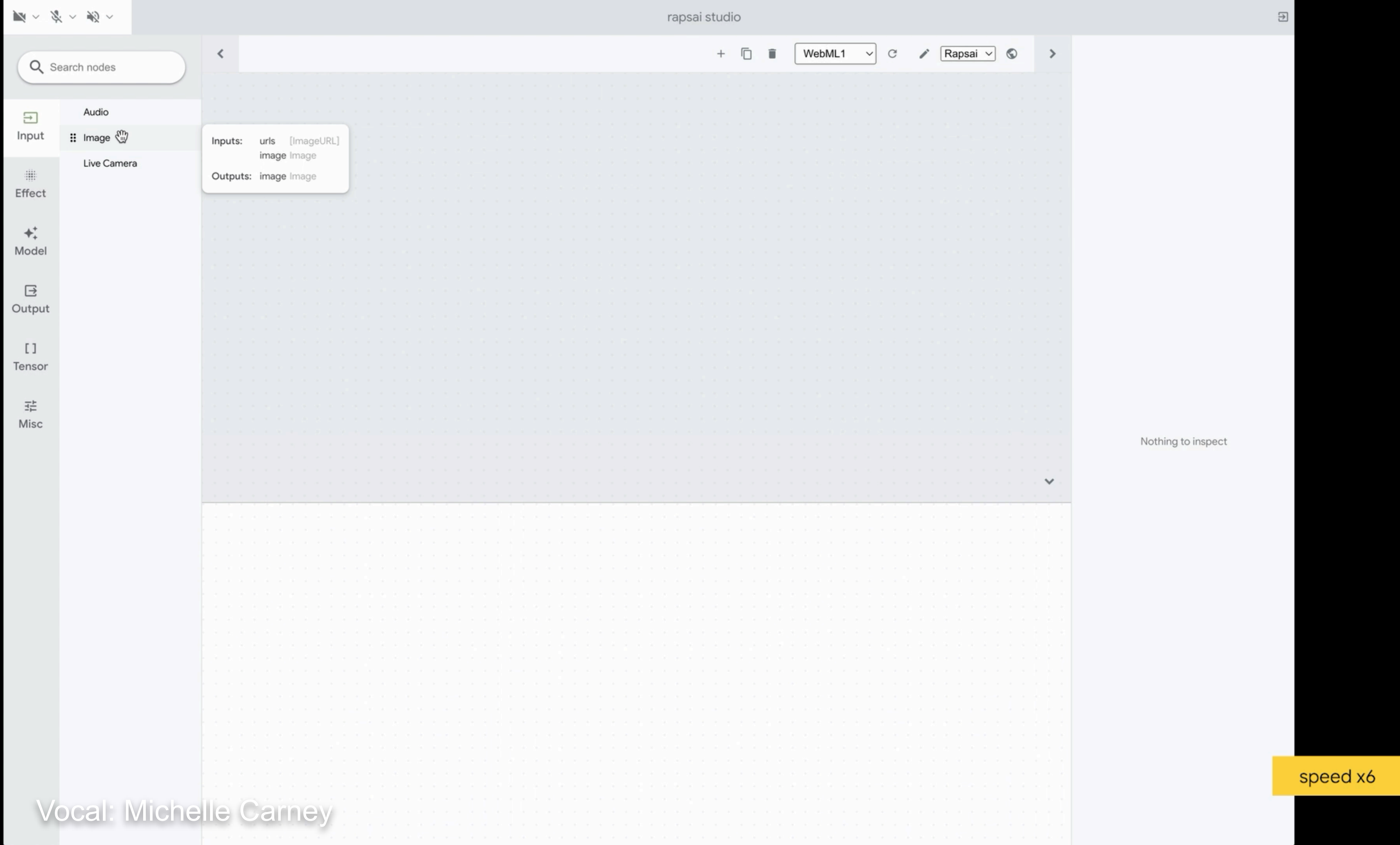
"We need to integrate the model with other modules — (to evaluate whether) can we improve the higher-level model?" (I7)

Design goals

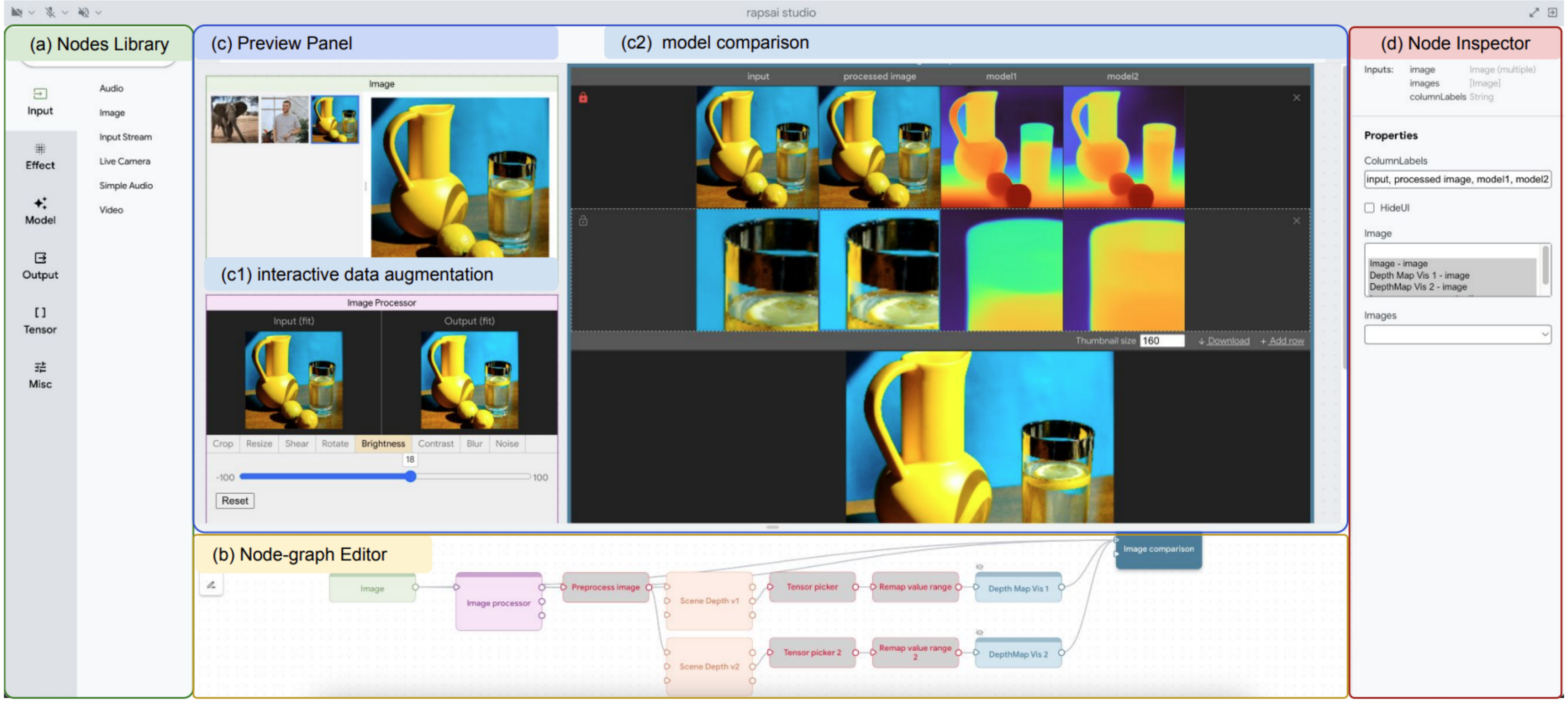
1. Visual programming for rapid prototyping
2. Run real-time ML pipelines
3. Input in-the-wild
4. Interactive data augmentation
5. Side-by-side comparison
6. Off-the-shelf & customize models

rapsai → Visual Blocks for ML

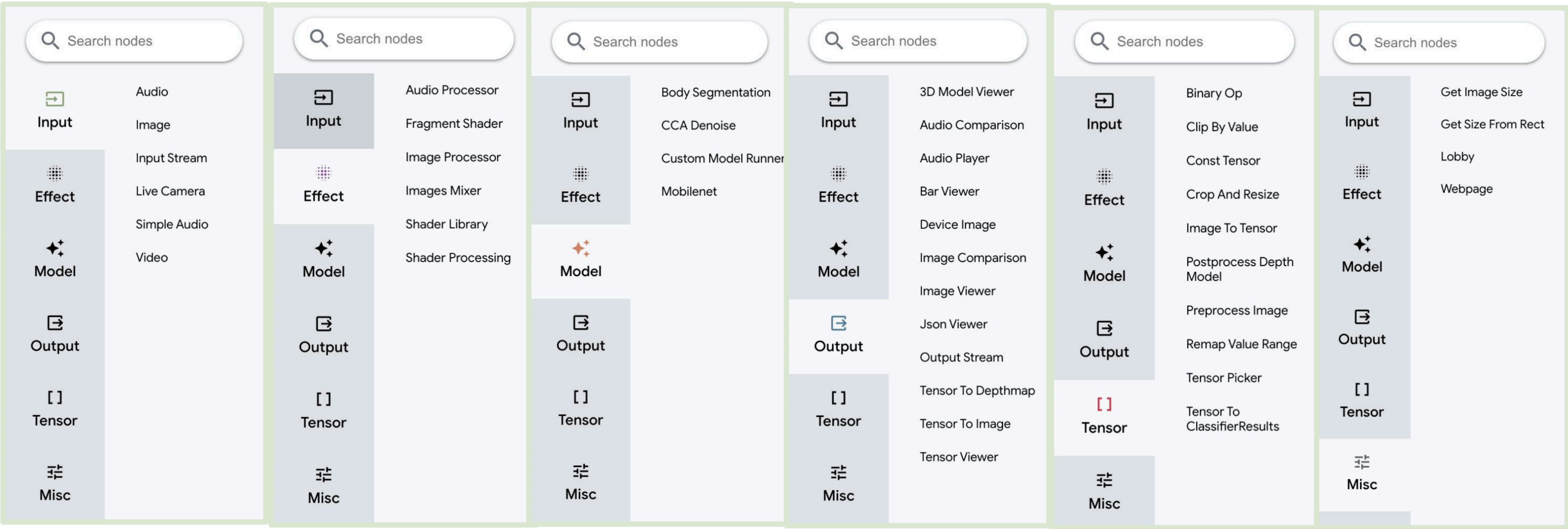
A Visual Programming Platform for Rapid and Iterative Development of End-to-end ML-based Applications


rapsai studio
Search nodes
Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera
Inputs: urls [ImageURL]
image Image
Outputs: image Image
WebML1
Rapsai
Nothing to inspect
speed x6
Vocal: Michelle Carney

Rapsai
System Overview



Rapsai
Nodes Library



Rapsai
Nodes Library

D FULL LIST OF NODES IN RAPSAI

We list all the supported nodes in Rapsai before the case study. Note that Rapsai is extendable for expert users by adding new nodes in JavaScript.

D.1 Input nodes

(1) **image**: capture a photo from webcam, upload from hard drive, or fetch from a list of remote URLs.
(2) **video**: record a video with external webcam or upload a video from disk or YouTube.
(3) **audio**: record sounds from microphone, or upload audio files from disk or Internet
(4) **live camera**: use live camera stream, similar for the live audio node.
(5) **remote stream**: stream input from another device (*e.g.*, mobile phone) via WebRTC, by opening a URL of the page with a lobby node.
(6) **lobby**: create a WebRTC server to accept video streaming from other devices. Remote stream nodes that connect to the lobby node with the same name are sharing output via WebRTC streaming.

D.2 Effect nodes

(7) **image processor**: crop and translate a region of interest in the input to verify an image model's invariance to translation; rotate, shear, resize an image to examine potential biasing issues in the training sets; apply blur and noise to test a model's robustness.
(8) **image mixer**: mix images with GPU-based blending modes[10]
(9) **audio processor**: trim the audio, change volume, and add background noise from a collection of 17 presets.
(10) **fragment shader**: program and apply a screen-space graphical shader effect.
(11) **shader library**: offers a pre-defined list of shader code to avoid coding into the "fragment shader".

D.3 Model nodes

(12) **custom model runner**: enter the URL of an TensorFlow.js model or upload a TensorFlow model into the pipeline.
(13) **body segmentation**: run a deployed MediaPipe body segmentation model.
(14) **audio denoising**: run either of two deployed audio denoising models.
(15) **MobileNet**: run a deployed MobileNet model for image classification.

D.4 Output nodes

(16) **image viewer** node displays an image.
(17) **audio player**: play a single audio output.
(18) **image comparison**: qualitatively compare output from multiple models with zoom-in tools.
(19) **audio comparison**: qualitatively compare output from multiple models with automatic track switching.
(20) **JSON viewer**: read the raw output from a model for debugging.
(21) **bar viewer**: view the classification results from a model such as MobileNet.
(22) **3D model viewer**: view 3D models from an URL or tensor.
(23) **tensor to image**: view a tensor as images.
(24) **tensor to depthmap**: view a tensor as depthmaps with different transfer functions.
(25) **output stream**: see remote video streams from the **lobby node** input.

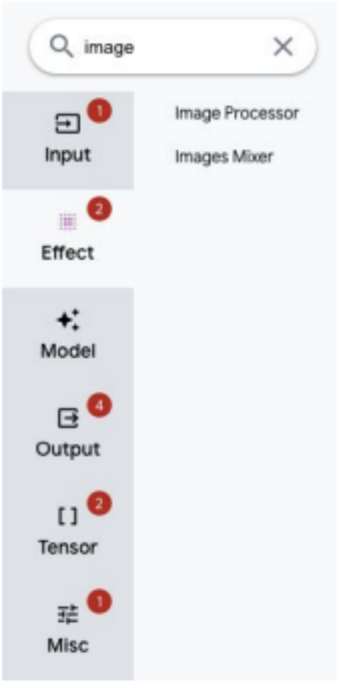
D.5 Tensor nodes

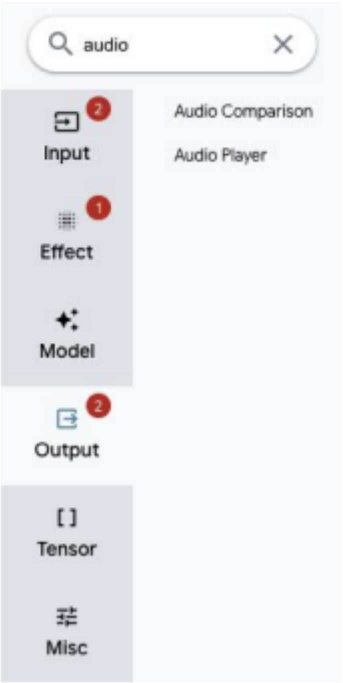
(26) **preprocess image**: converts an input image to a 4D tensor as an input that is required by most image models.
(27) **tensor picker**: select a tensor from an array of output tensors.
(28) **tensor postprocess**: convert a tensor to an image and apply normalization calculators.
(29) **binary op**: apply "and", "or", "xor", and "not" operations between two input tensors.
(30) **clip by value**: clamp the values of an input tensor.
(31) **crop and resize**: crop and resize a two-dimensional tensor.
(32) **preprocess tensor**: normalize a tensor, expand dimensions, and optionally convert to grayscale image for many generative models.
(33) **postprocess tensor**: normalize and resize a tensor for image output.
(34) **tensor picker**: select a certain tensor from an array of model output. Note that most models only have one output so it is optional.
(35) **remap value range**: select an input and an output ranges to remap tensor values.

D.6 Miscellaneous nodes

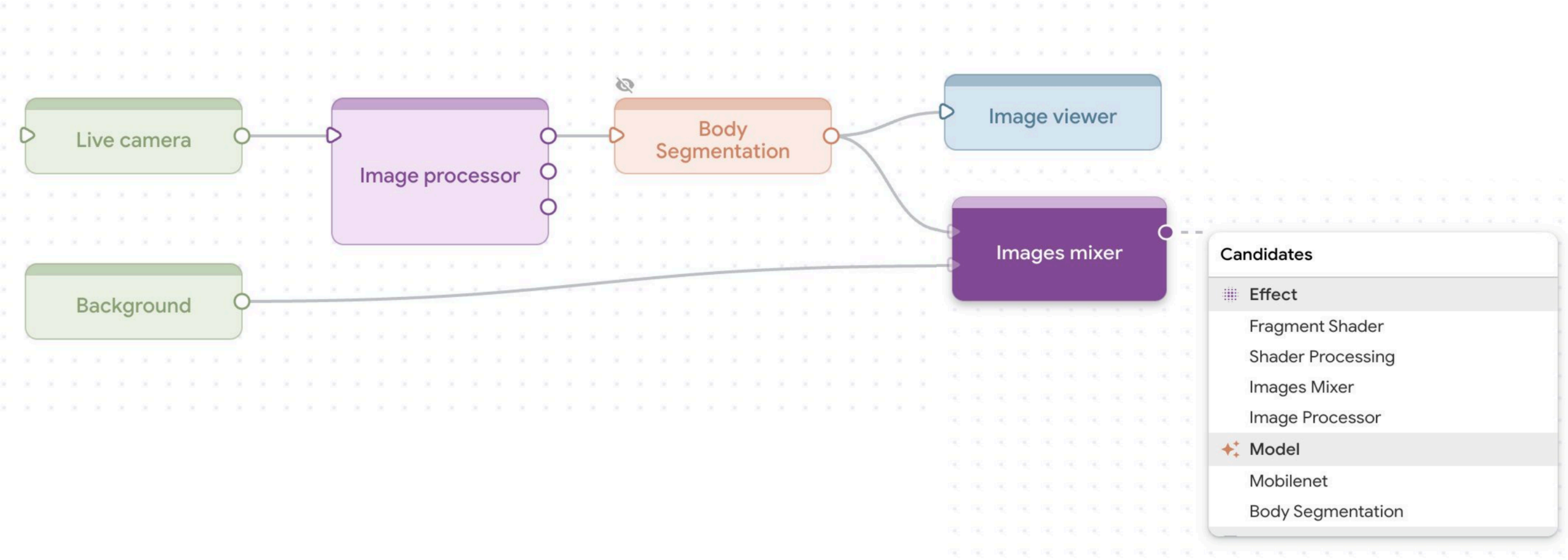
(36) **webpage**: append a Google Form or a custom webpage for filling in surveys.
(37) **image size**: obtain image size for outputting to some models.

Rapsai
Nodes Library

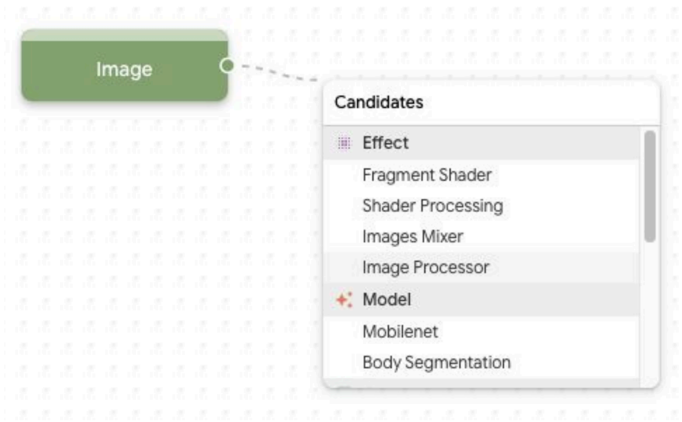




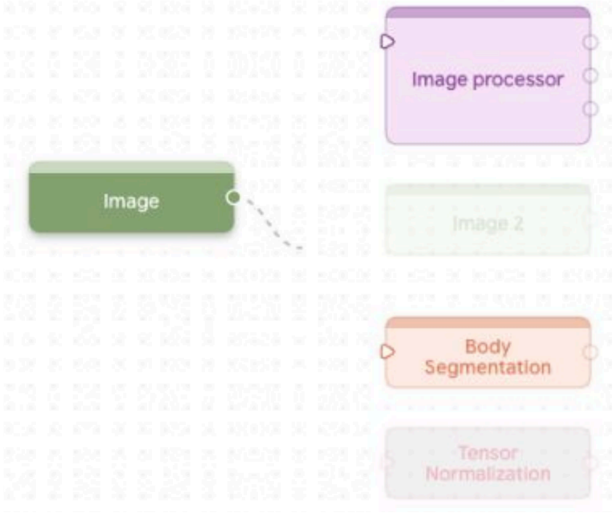
Rapsai
Node-graph Editor



Rapsai
Node-graph Editor

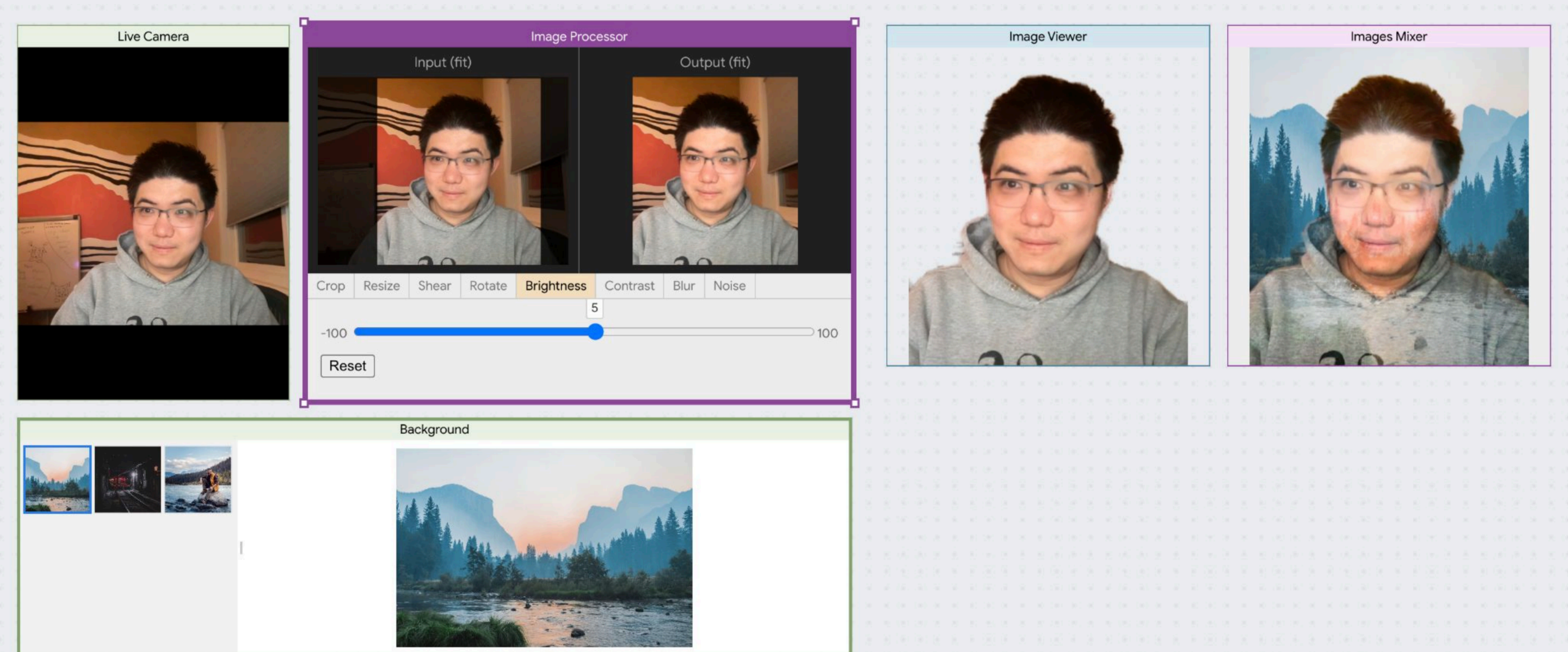


Node Suggestion



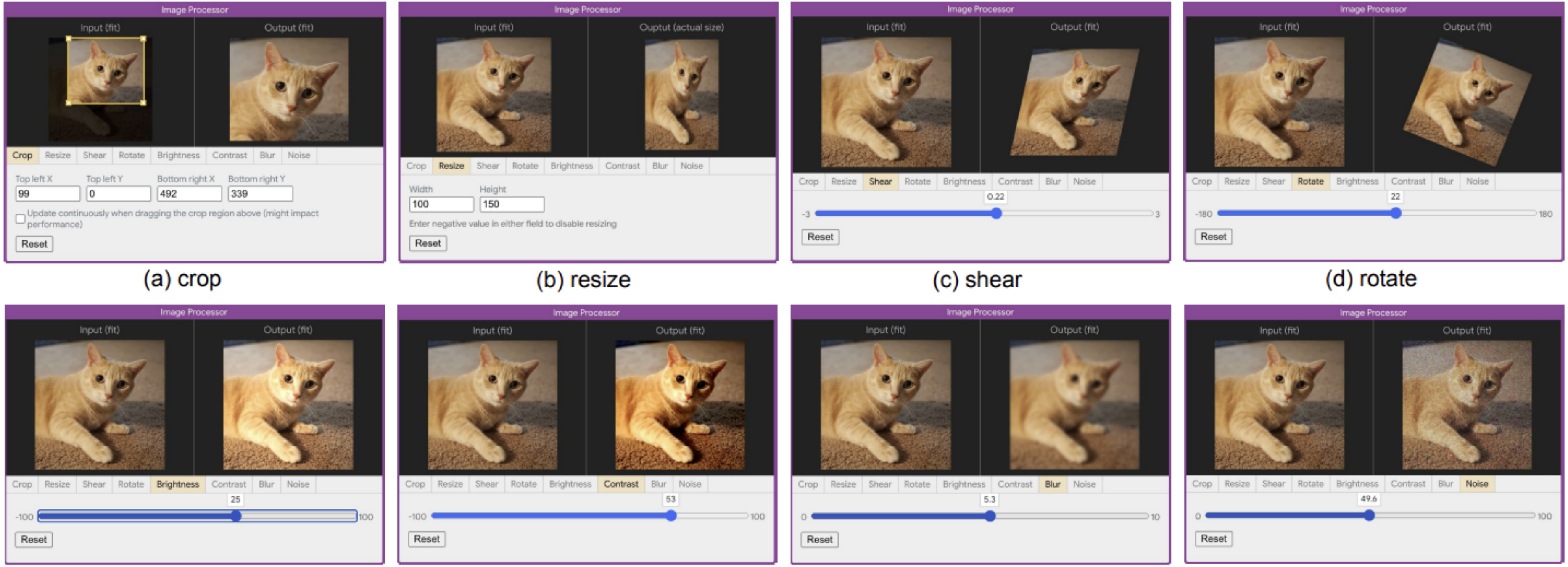
Link Suggestion

Rapsai
Preview Panel



Rapsai

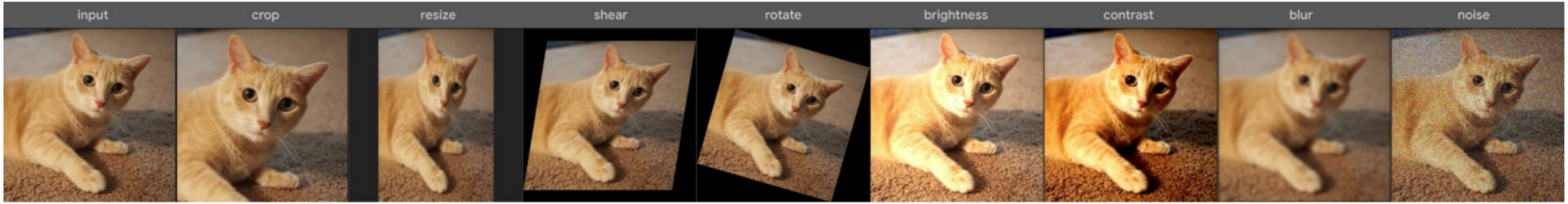
Interactive Data Augmentation



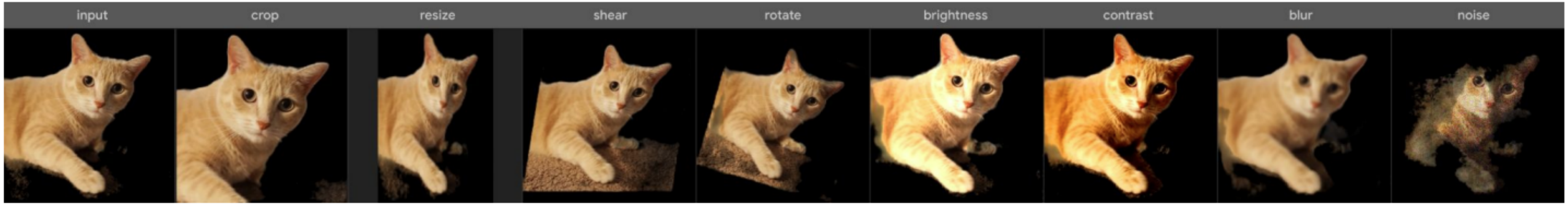
(a) crop (b) resize (c) shear (d) rotate

(e) brightness (f) contrast (g) blur (h) noise

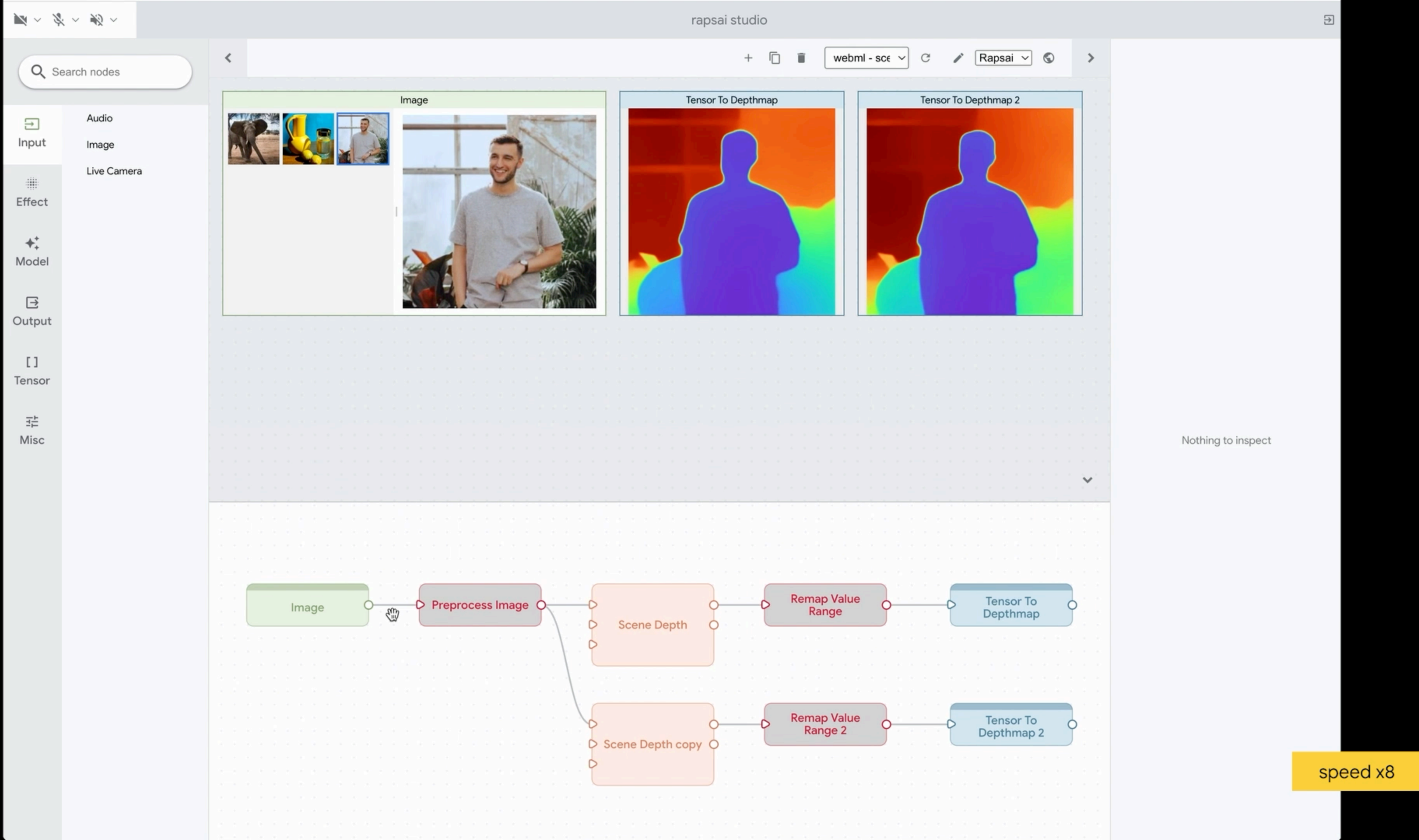
Rapsai
Interactive Data Augmentation



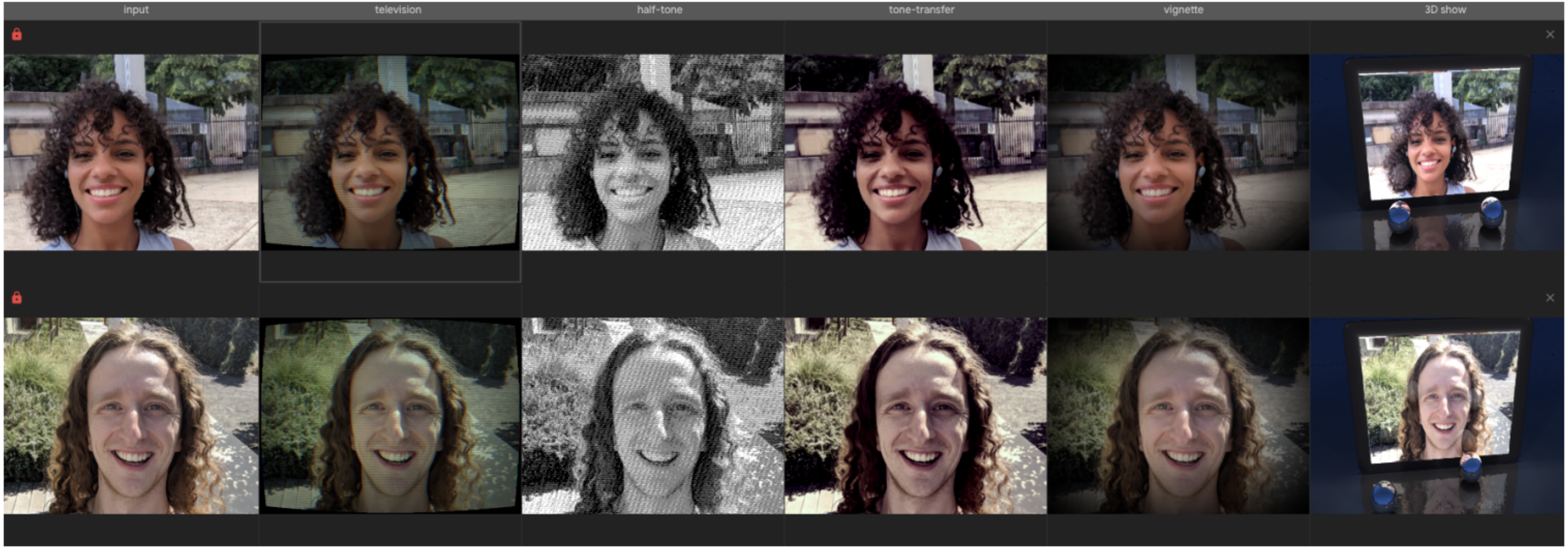
(a) intermediate results of interactive data augmentation applied onto a single input image

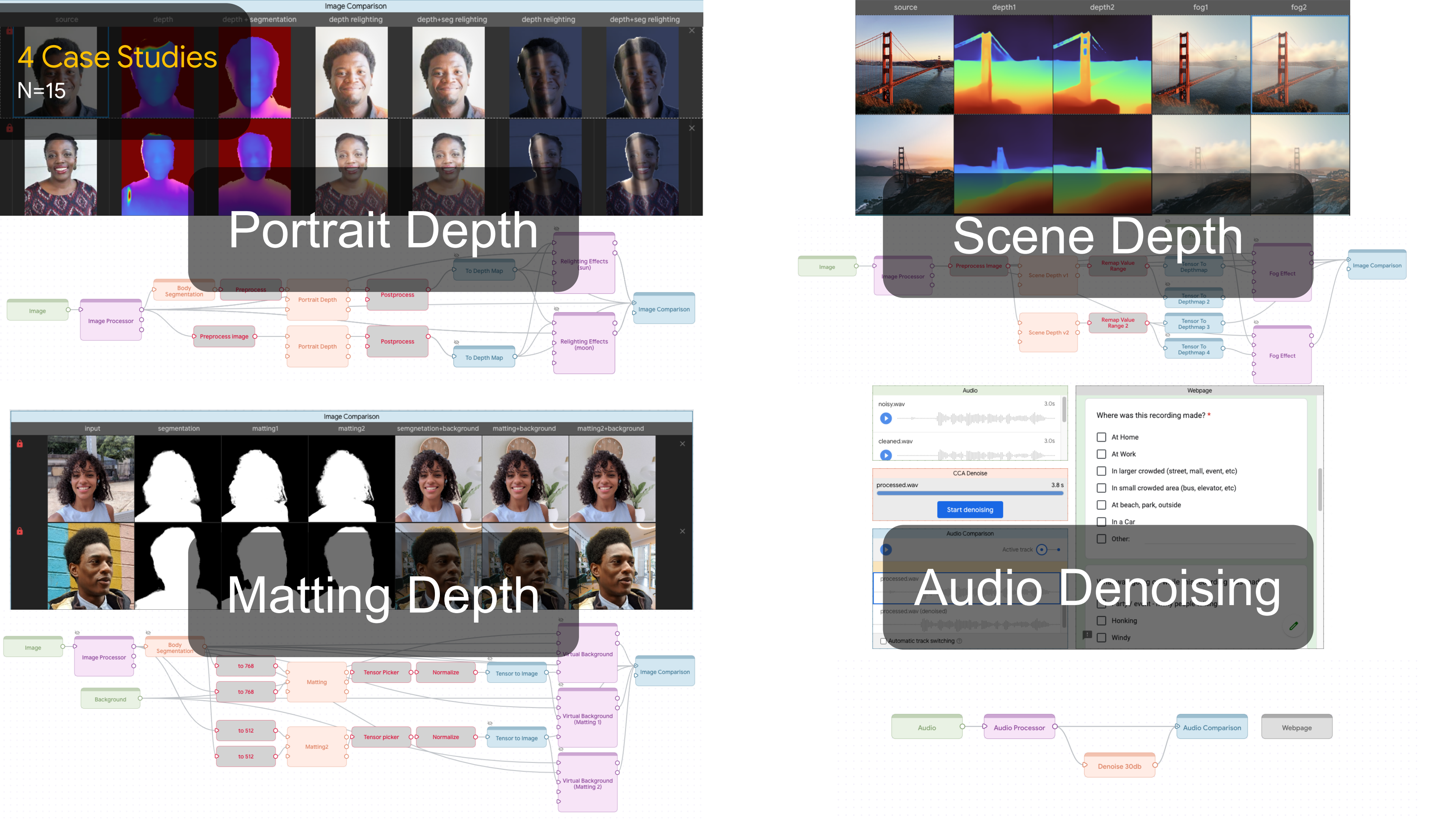


(b) side-by-side comparison of segmentation results with different augmentation techniques


rapsai studio
Search nodes
Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera
webml - sce
Rapsai
Image
Tensor To Depthmap
Tensor To Depthmap 2
Nothing to inspect
Image
Preprocess Image
Scene Depth
Remap Value Range
Tensor To Depthmap
Scene Depth copy
Remap Value Range 2
Tensor To Depthmap 2
speed x8

Rapsai
Shader Library

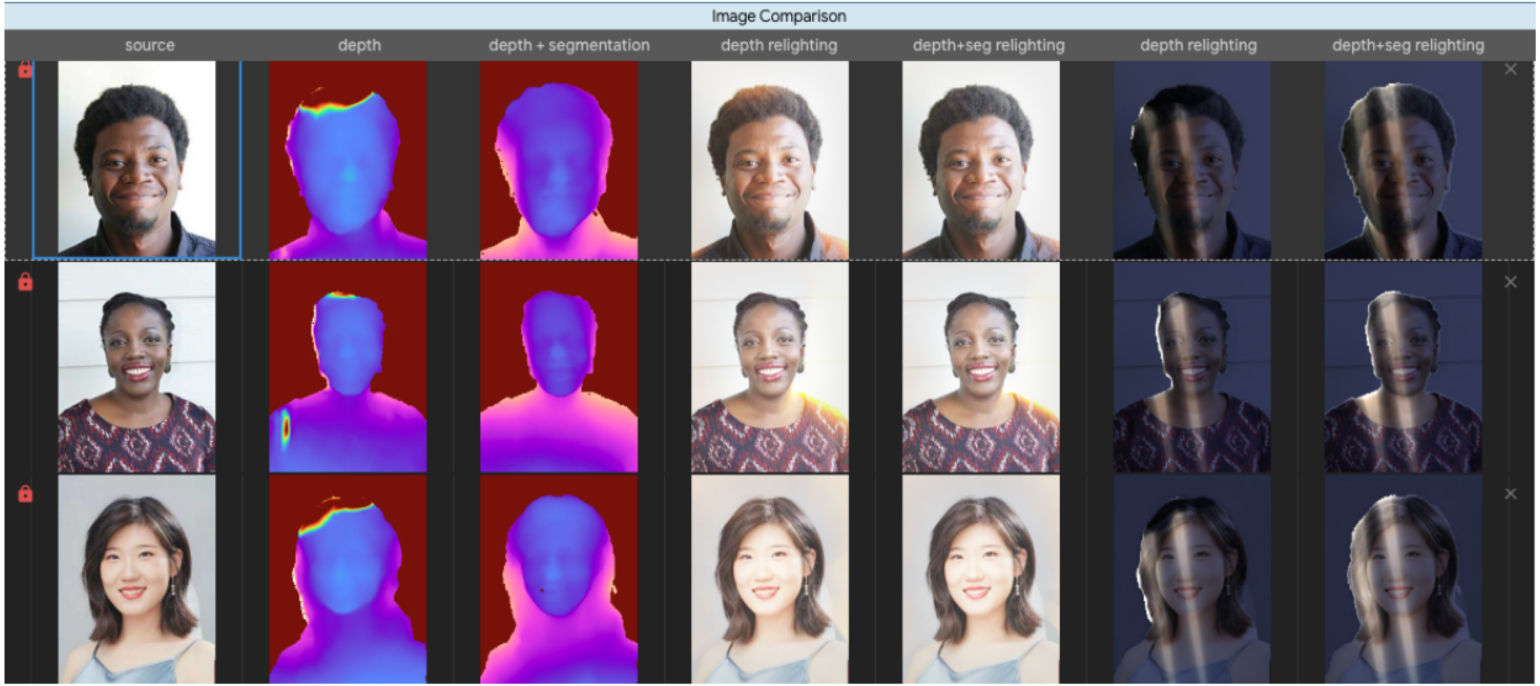


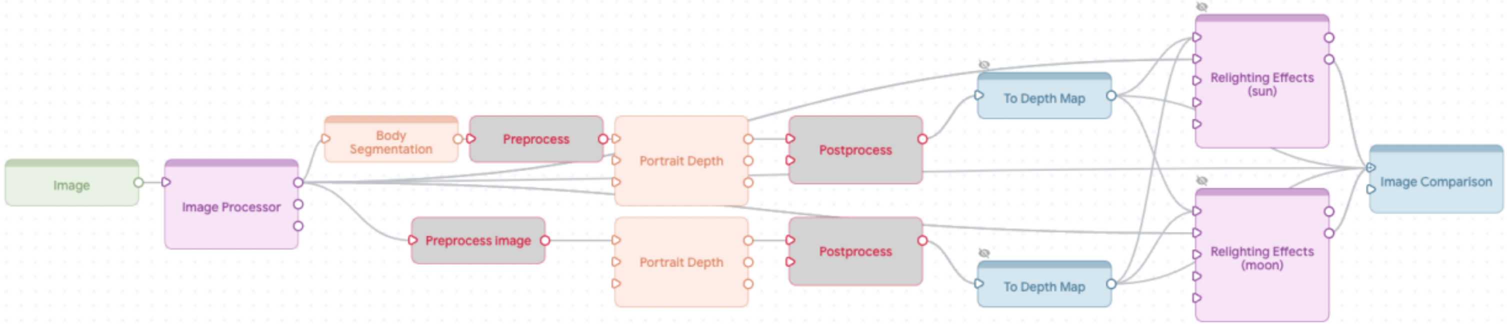


4 Case Studies
N=15
Portrait Depth
Image Comparison
source
depth
depth + segmentation
depth relighting
depth+seg relighting
depth relighting
depth+seg relighting
Image
Image Processor
Body Segmentation
Preprocess
Portrait Depth
Postprocess
To Depth Map
Relighting Effects (sun)
Preprocess image
Portrait Depth
Postprocess
To Depth Map
Relighting Effects (moon)
Image Comparison
Scene Depth
source
depth1
depth2
fog1
fog2
Image
Image Processor
Preprocess Image
Scene Depth v1
Remap Value Range
Tensor To Depthmap
Tensor To Depthmap 2
Fog Effect
Image Comparison
Scene Depth v2
Remap Value Range 2
Tensor To Depthmap 3
Tensor To Depthmap 4
Fog Effect
Matting Depth
Image Comparison
input
segmentation
matting1
matting2
semgnetation+background
matting+background
matting2+background
Image
Image Processor
Body Segmentation
Background
to 768
to 768
Matting
Tensor Picker
Normalize
Tensor to Image
Virtual Background
to 512
to 512
Matting2
Tensor picker
Normalize
Tensor to Image
Virtual Background (Matting 1)
Virtual Background (Matting 2)
Image Comparison
Audio Denoising
Audio
noisy.wav
3.0s
cleaned.wav
3.0s
CCA Denoise
processed.wav
3.8 s
Start denoising
Audio Comparison
Active track
processed.wav
processed.wav (denoised)
Automatic track switching
Webpage
Where was this recording made? *
At Home
At Work
In larger crowded (street, mall, event, etc)
In small crowded area (bus, elevator, etc)
At beach, park, outside
In a Car
Other:
Honking
Windy
Audio
Audio Processor
Denoise 30db
Audio Comparison
Webpage

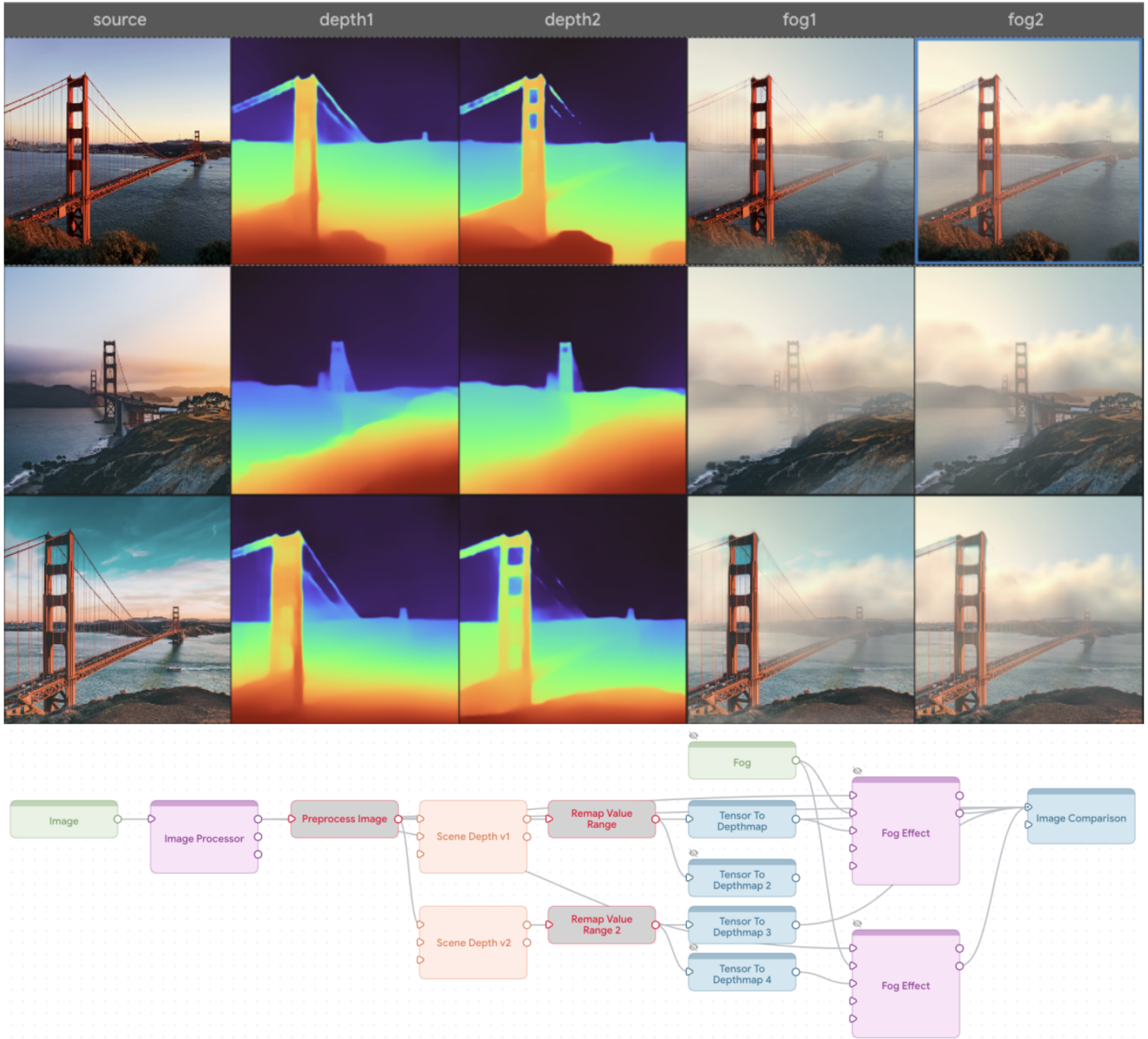
Case Study 1

Portrait Depth

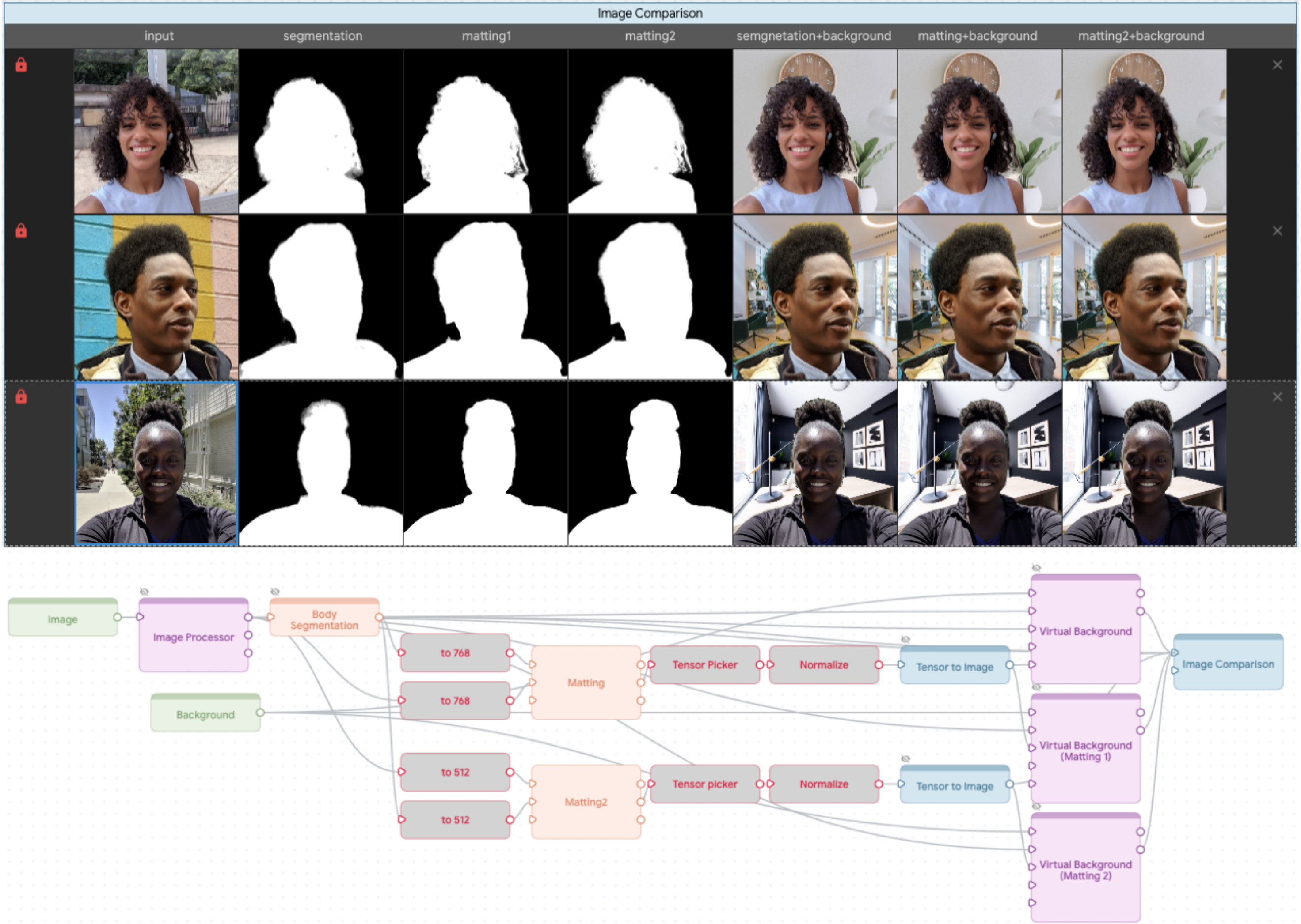




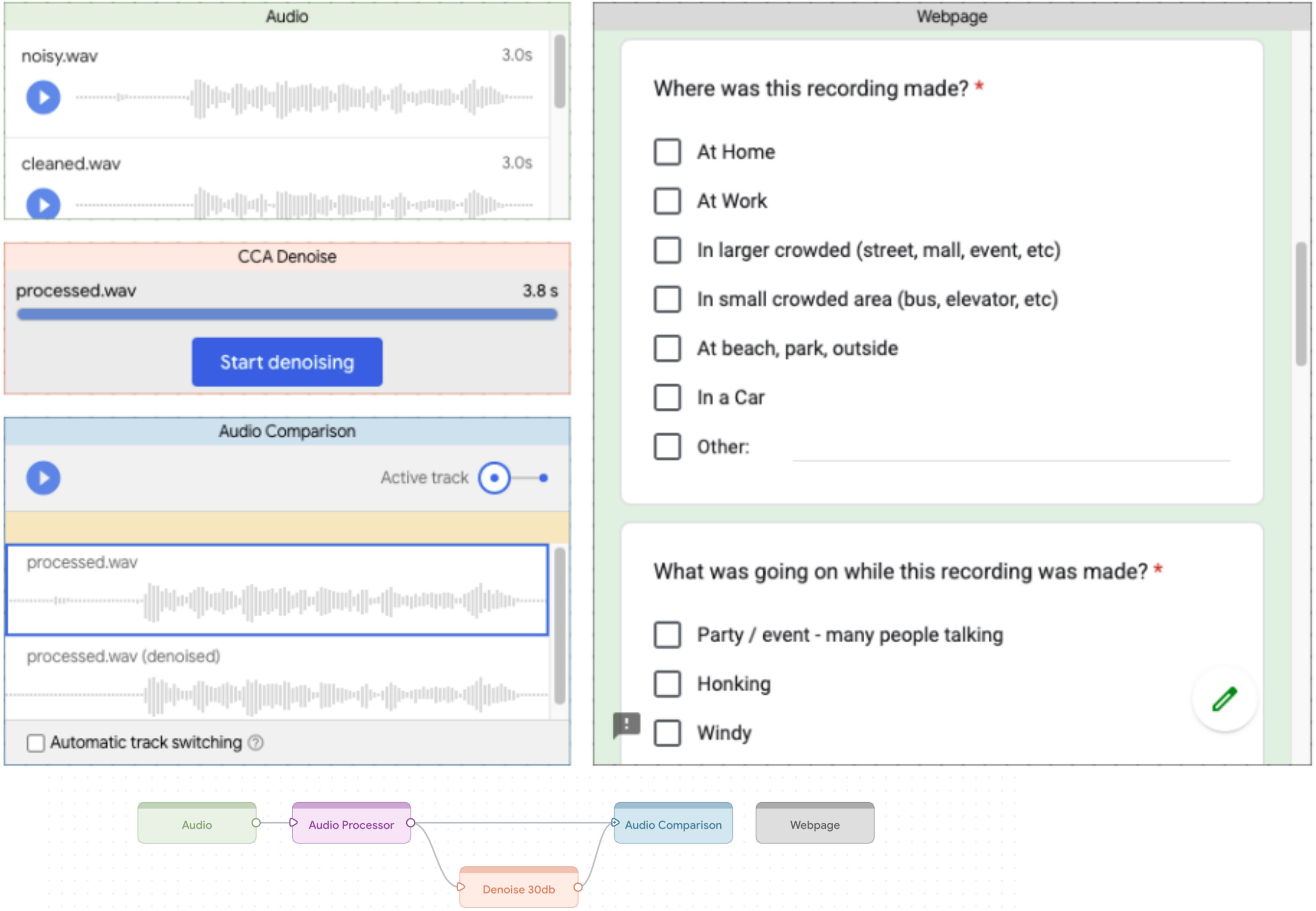
Case Study 2
Scene Depth



Case Study 3
Alpha Matte



Case Study 4
Audio Denoising



Case Study

Five Stages

Background interview (6.1 ± 0.8 min)

Video tutorial (4 min),

Visual analytics procedure using Rapsai (39.4 ± 4.6 min)

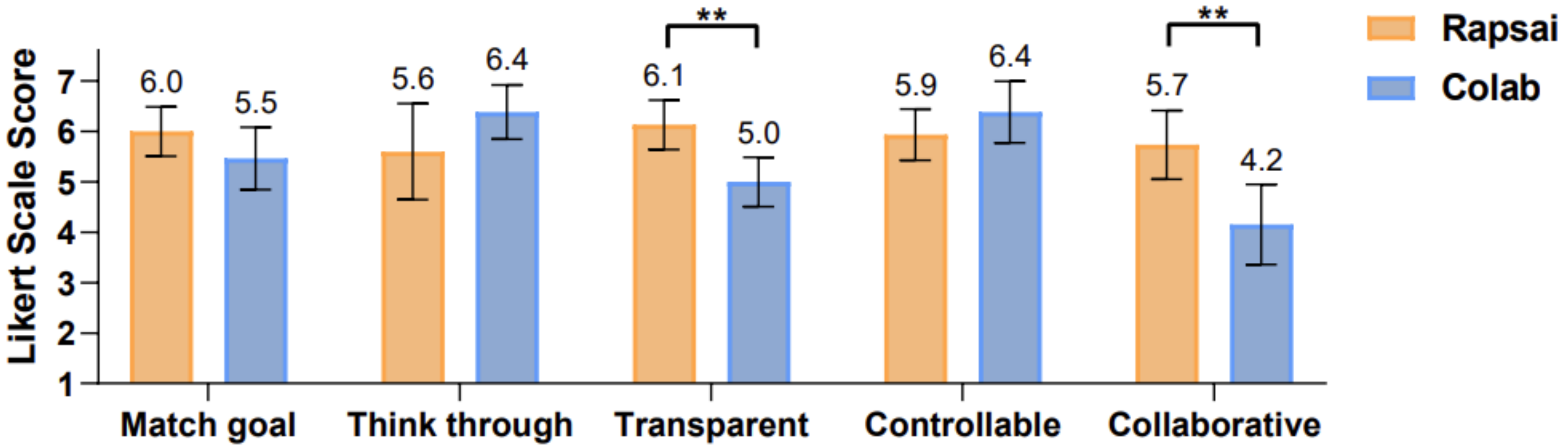
Discussion of Rapsai and perception prototyping in future (10.2 ± 2.0 min)

Post-hoc exit survey to use Rapsai and compare with Colab

Findings 1

Rapsai *vs* Colab

Less Control but *More Transparent* and *Collaborative*



I spent about half a minute to create an image classification pipeline, and I spent 2–3 minutes to build a depth estimation pipeline from scratch, since it took some time to figure out how to preprocess the input and visualize the output... while Colab is more flexible for different tasks, I guess it could range from 1 hour to a day or two.

P6

“

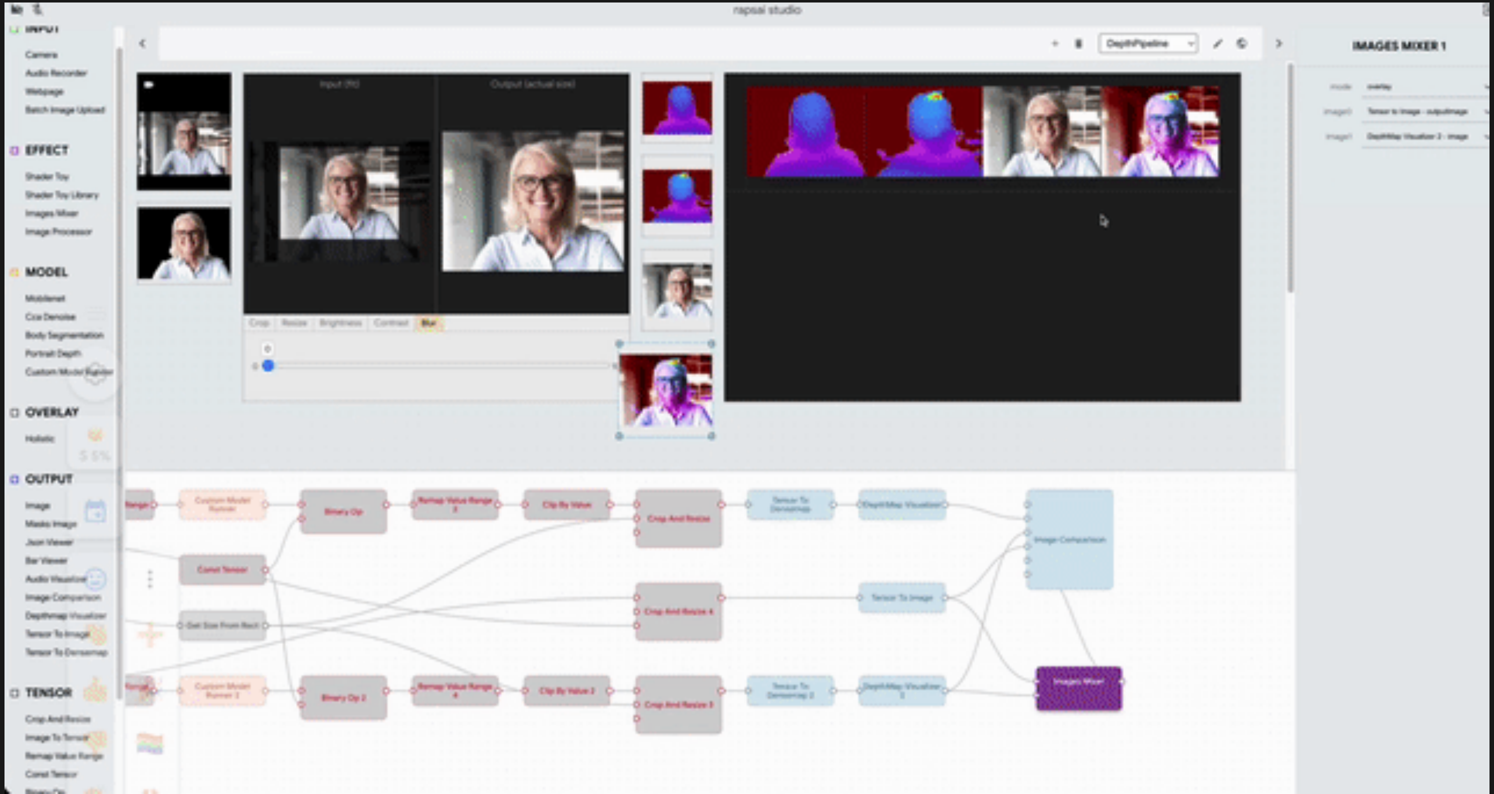
In my case, I started from an existing template but overall it was quite fast, I’d say less than 5 min.

”

P13

Findings 2

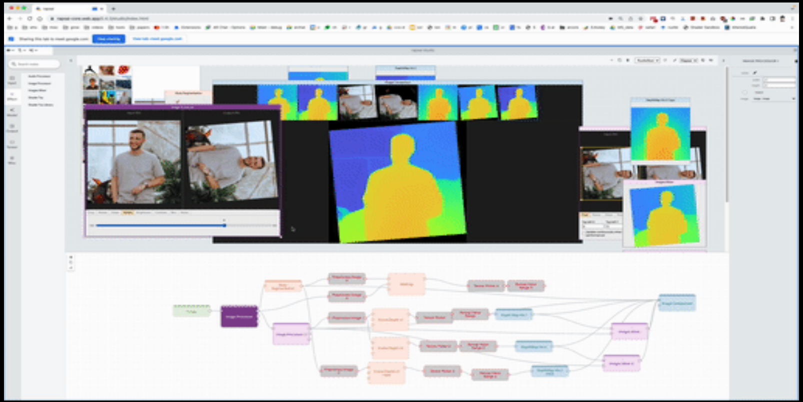
Assist in *Identifying Issues* with ML Models and Training Sets



“

It can help me understand how I should change the model architecture and what training examples to add.

”

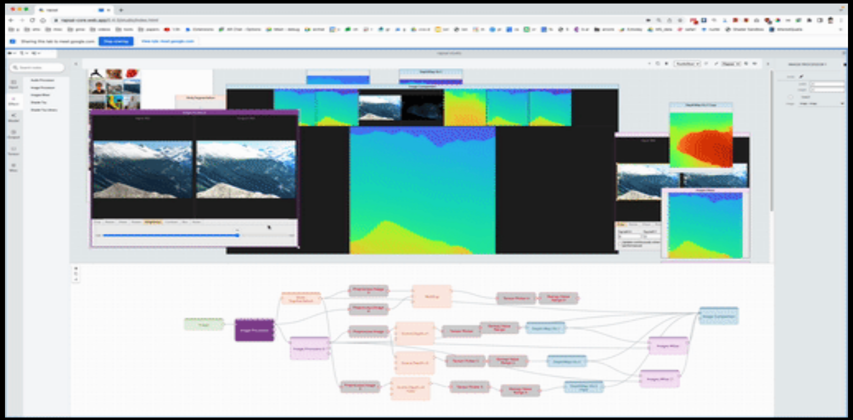


P10

“

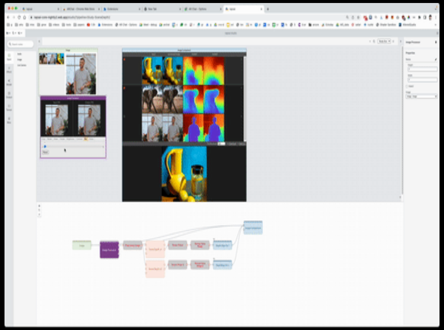
I can manipulate the brightness to see when the model fails.

”



P2

“It gives me an intuition about which data augmentation operations that my model is more sensitive, then I can go back to my training pipeline, maybe increase the amount of data augmentation for those specific steps that are making my model more sensitive.”

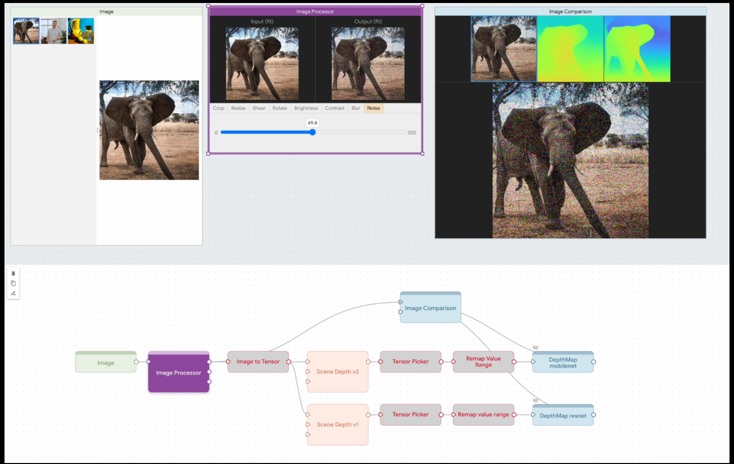


P3

Using a video <as input> helps me get a cross-time feel of how the model performance varies, which is hard to capture with metrics.

P10

Comparing various noise parameters in the input to a model is useful to identify augmentation bias.

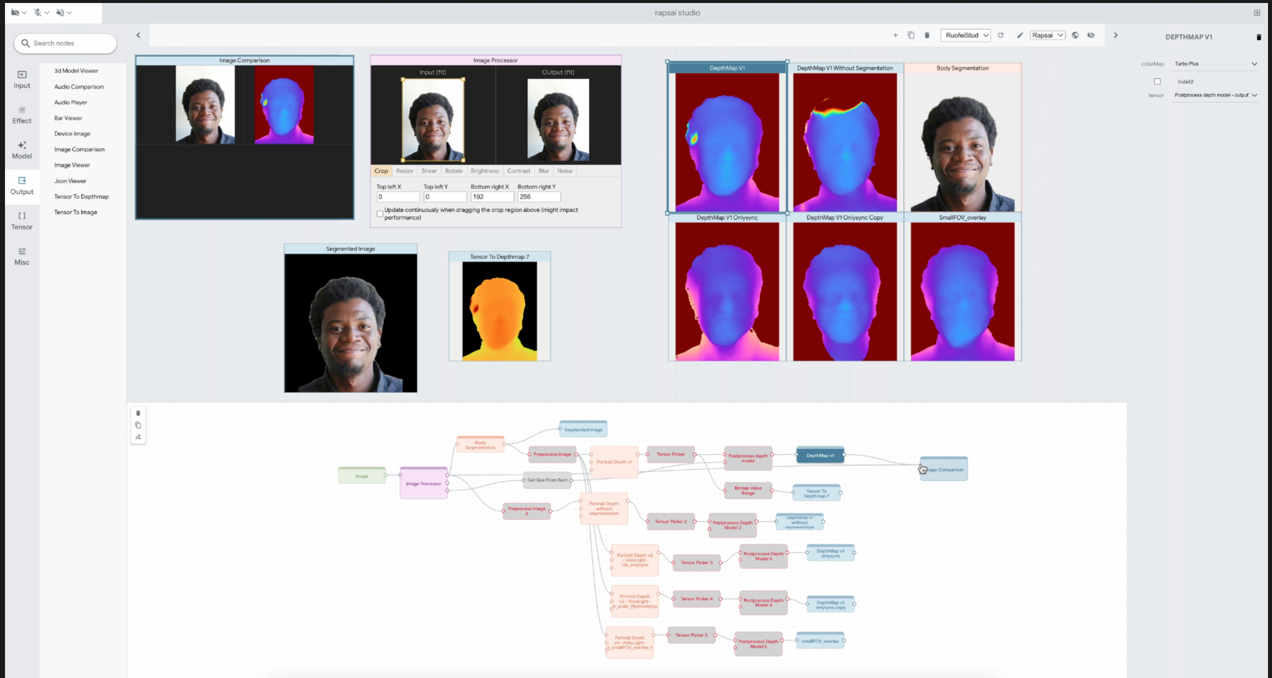


P8

Findings 3

Rapsai helps

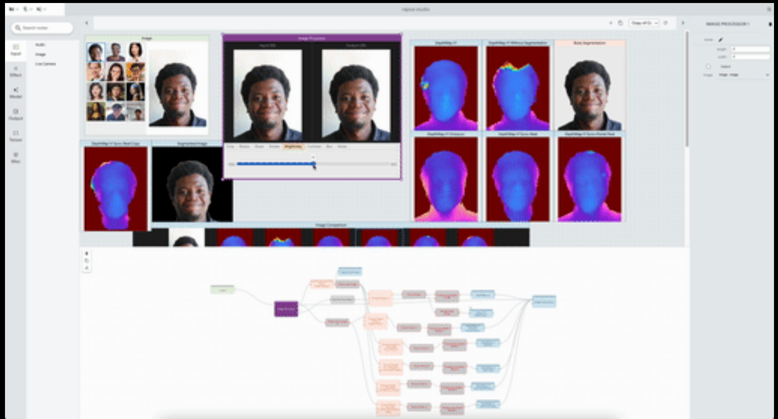
Model Selection,
Learning From Pipelines,
Study deployment



Building a custom webpage as debugging tool [by coding], cost <a junior engineer> over a month to build. This [Rapsai] is easy to distribute and try it immediately. It helps debug the pipeline.

P9

“It can help me understand how should I change the model architecture and add what training examples.”



P1

Visual Blocks for ML

rapsai

1. Lowers the barriers for ML prototyping

2. Empowers users to experiment with no/low-code environment

3. Facilitates collaboration between designers and developers

BLOG ›

Visual Blocks for ML: Accelerating machine learning prototyping with interactive tools

FRIDAY, APRIL 21, 2023

Posted by Ruofei Du, Interactive Perception & Graphics Lead, Google Augmented Reality, and Na Li, Tech Lead Manager, Google CoreML

Recent deep learning advances have enabled a plethora of high-performance, real-time multimedia applications based on machine learning (ML), such as human body segmentation for video and teleconferencing, depth estimation for 3D reconstruction, hand and body tracking for interaction, and audio processing for remote communication.

However, developing and iterating on these ML-based multimedia prototypes can be challenging and costly. It usually involves a cross-functional team of ML practitioners who fine-tune the models, evaluate robustness, characterize strengths and weaknesses, inspect performance in the end-use context, and develop the applications. Moreover, models are frequently updated and require repeated integration efforts before evaluation can occur, which makes the workflow ill-suited to design and experiment.

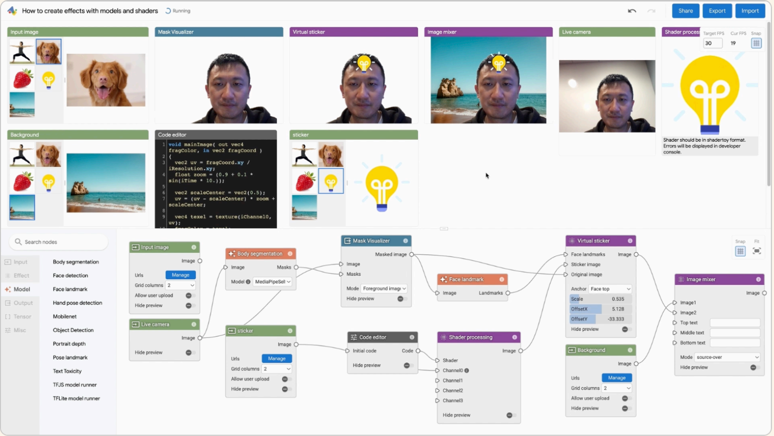
In "Rapsai: Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming", presented at CHI 2023, we describe a visual programming platform for rapid and iterative development of end-to-end ML-based multimedia applications. Visual Blocks for ML, formerly called Rapsai, provides a no-code graph building experience through its node-graph editor. Users can create and connect different components (nodes) to rapidly build an ML pipeline, and see the results in real-time without writing any code. We demonstrate how this platform enables a better model evaluation experience through interactive characterization and visualization of ML model performance and interactive data augmentation and comparison. Sign up to be notified when Visual Blocks for ML is publicly available.

Visual Blocks for ML

Unleash your creativity

Visual Blocks for ML is an experimental JavaScript framework from Google that helps you add drag-and-drop machine learning blocks to your platform. Only your imagination limits the blocks you give your users. Off the shelf blocks include models, user inputs, processors and visualizations.

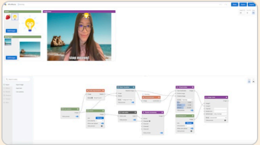
Colab experience and open source library will come soon.



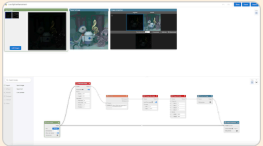
With Visual Blocks for ML, you can drag and drop to make your ML no coding required

Experiment with Visual Blocks

Click on any of the demos to pull up its graph in the interaction editor below



AR Effects



Low Light Enhancement

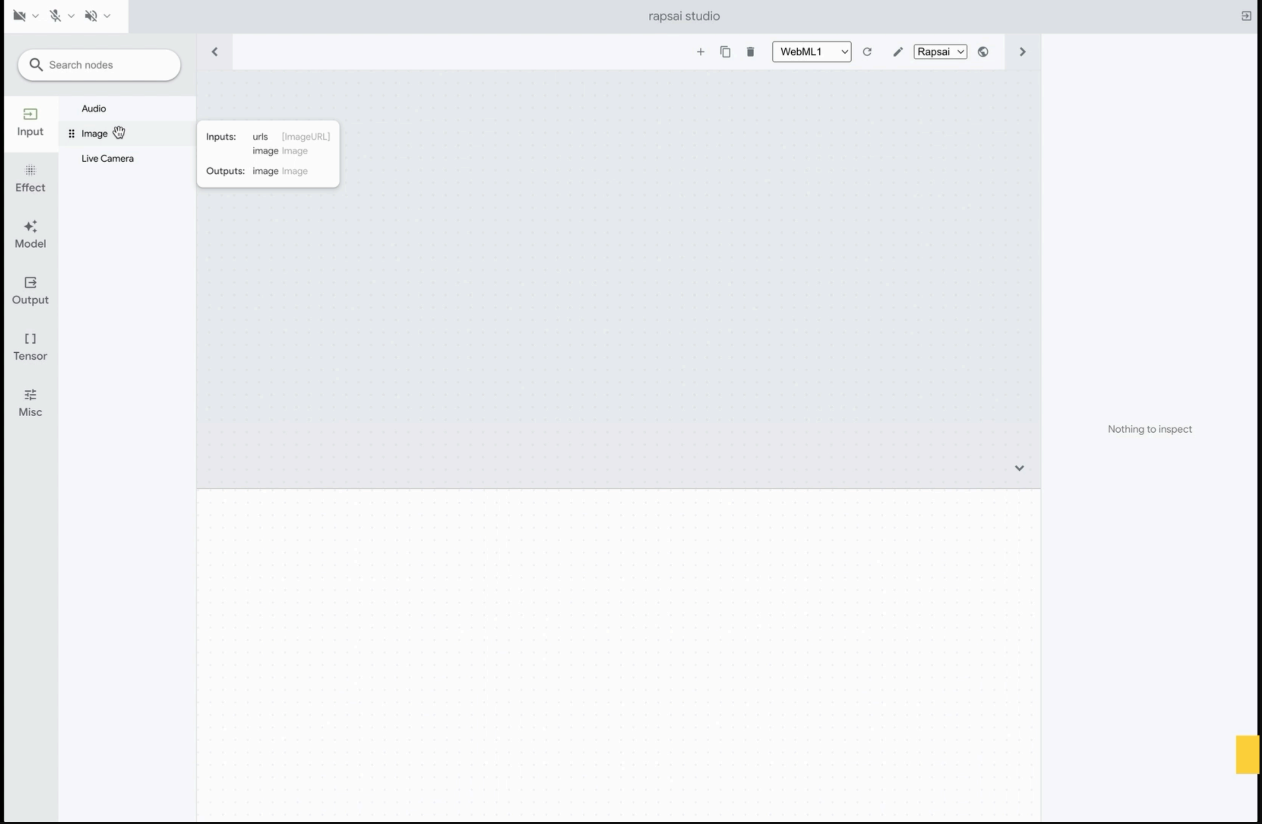
Create Your Own

rapsai

With the right tools,
everyone can unleash
your inner creativity.

yet another
20% project

Honorable
Mention



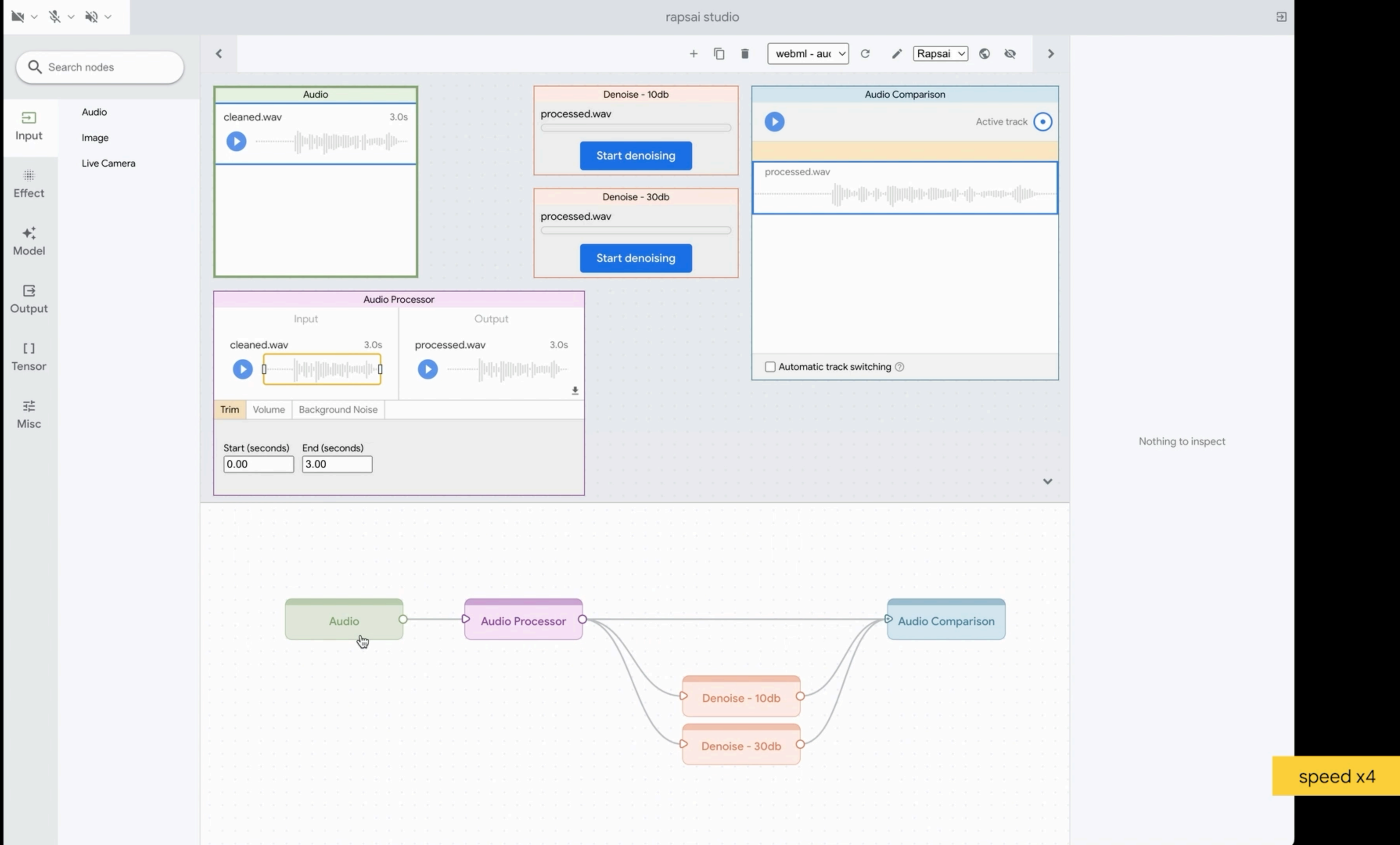
visualblocksforml.github.io

rapsai

Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu "Bruce" Liu, Ahmed Sabie, Sergio Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal

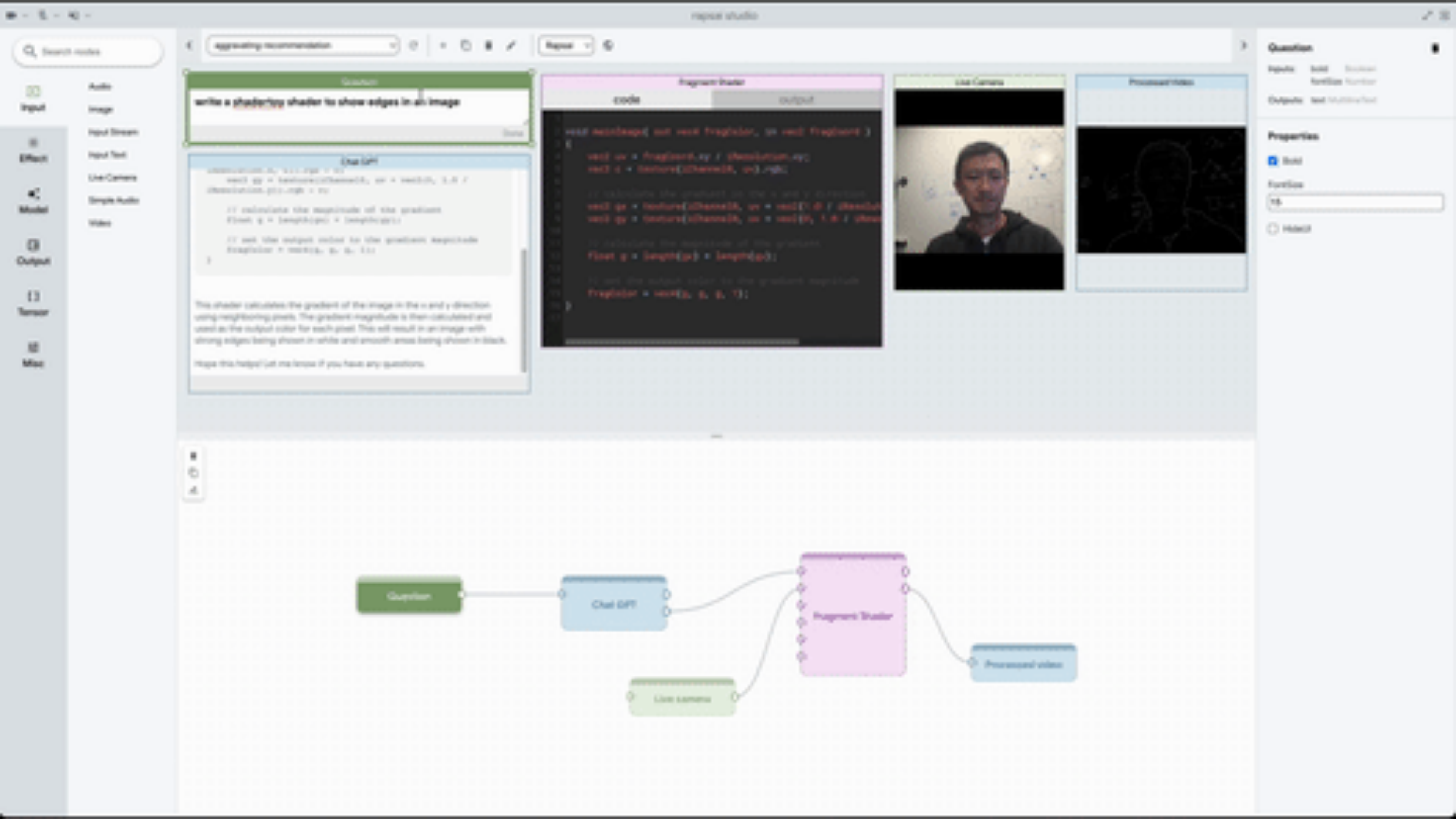
Google

CHI23
Hamburg, Germany | Hybrid
April 23-28, 2023
reCHInnecting

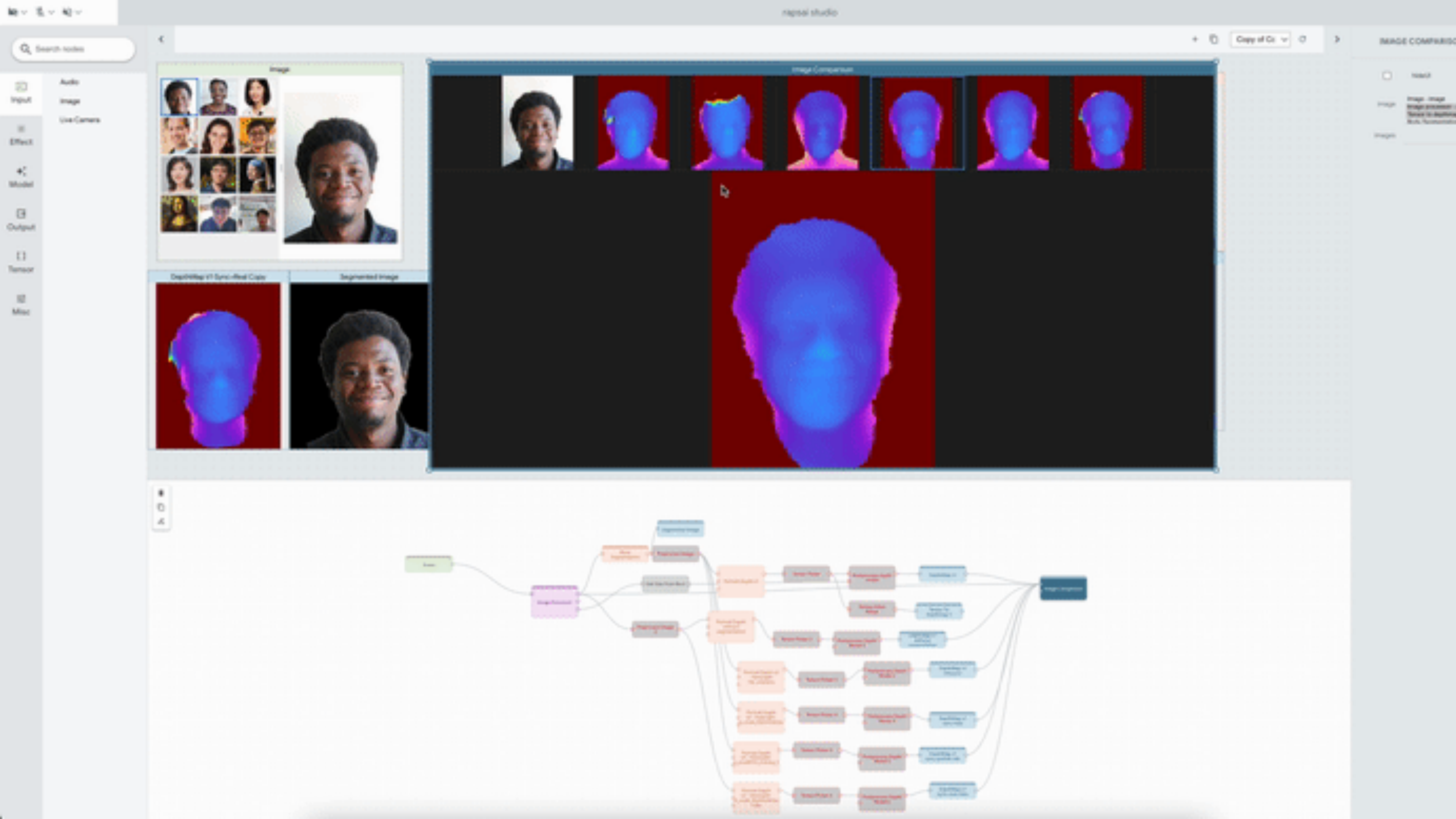

rapsai studio
Search nodes
Input
Audio
Image
Live Camera
Effect
Model
Output
Tensor
Misc
webml - auc
Rapsai
Audio
cleaned.wav
3.0s
Denoise - 10db
processed.wav
Start denoising
Denoise - 30db
processed.wav
Start denoising
Audio Comparison
Active track
processed.wav
Automatic track switching
Audio Processor
Input
Output
cleaned.wav
3.0s
processed.wav
3.0s
Trim
Volume
Background Noise
Start (seconds)
0.00
End (seconds)
3.00
Nothing to inspect
Audio
Audio Processor
Audio Comparison
Denoise - 10db
Denoise - 30db
speed x4

“There are some times when you don’t want all noise cancellation. People sometimes prefer audio with less noise cancellation because you want some context.”

P11



Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Input Stream
Input Text
Live Camera
Simple Audio
Video
Question
write a shadertoy shader to show edges in an image
Chat GPT
Fragment Shader
code
output
Live Camera
Processed Video
Properties
Bold
Question
Chat GPT
Fragment Shader
Live camera
Processed video

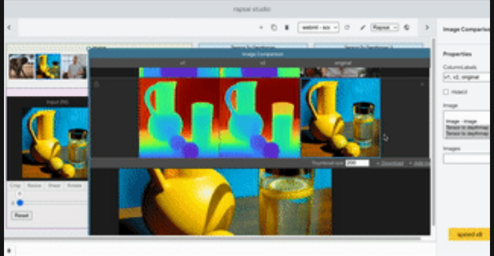

Search nodes
Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera
Image
Image Comparison
Segmented Image

yet another
20% project

Honorable
Mention

rapsai

Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming



Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu "Bruce" Liu, Ahmed Sabie, Sergio Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal

visualblocksforml.github.io

Google

CHI23
Hamburg, Germany | Hybrid
April 23-28, 2023
reCHInnecting

Honorable Mention

rapsai

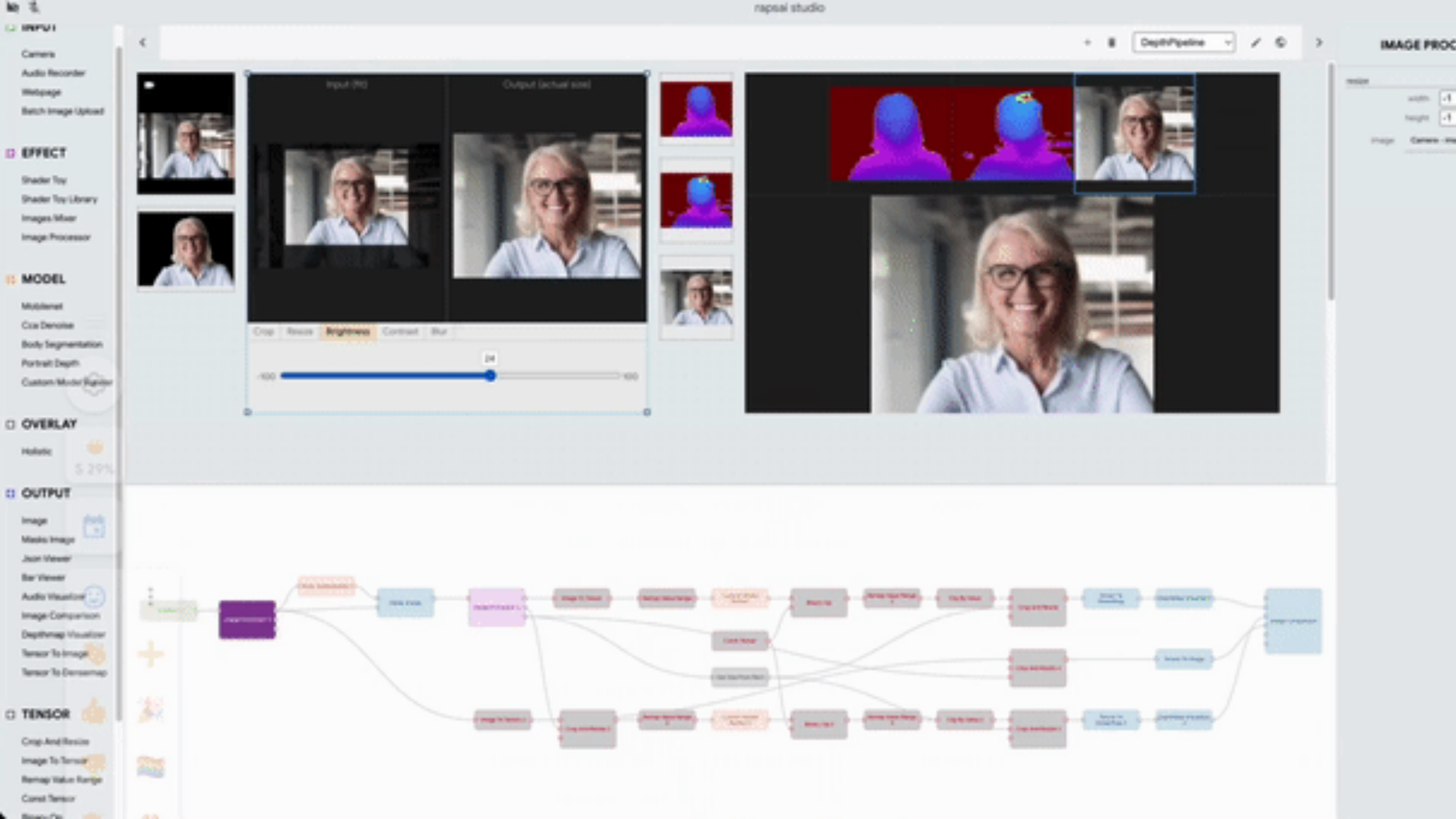
Accelerating Machine Learning Prototyping of Multimedia Applications through Visual Programming

Ruofei Du, Na Li, Jing Jin, Michelle Carney, Scott Miles, Maria Kleiner, Xiuxiu Yuan, Yinda Zhang, Anuva Kulkarni, Xingyu "Bruce" Liu, Ahmed Sabie, Sergio Escolano, Abhishek Kar, Ping Yu, Ram Iyengar, Adarsh Kowdle, and Alex Olwal

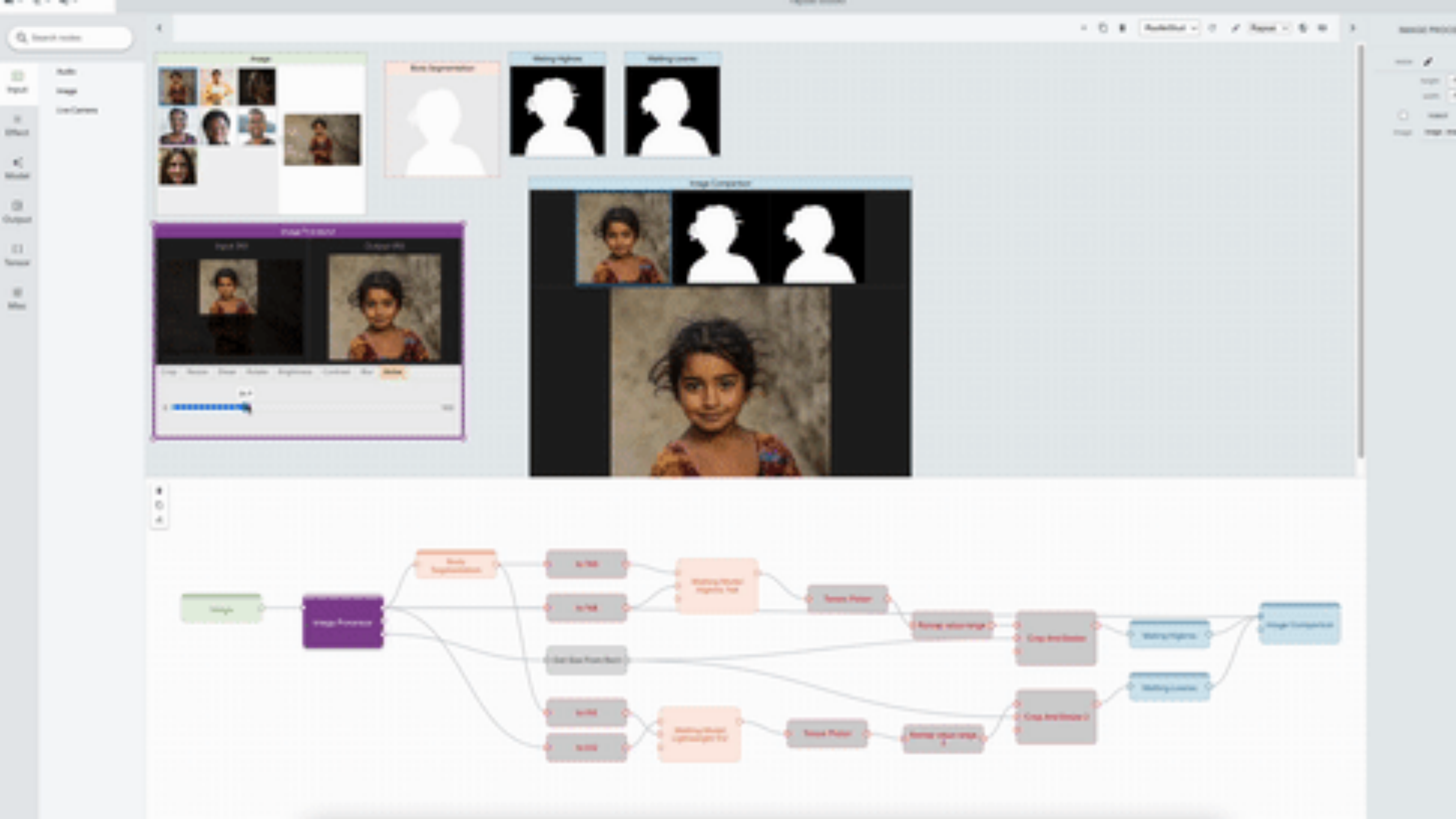
visualblocksforml.github.io

Google

CHI23
Hamburg, Germany | Hybrid
April 23-28, 2023
reCHInnecting



INPUT
Camera
Audio Recorder
Webpage
Batch Image Upload
EFFECT
Shader Toy
Shader Toy Library
Images Mixer
Image Processor
MODEL
Mobilenet
Body Segmentation
Portrait Depth
OVERLAY
Holistic
OUTPUT
Image
Masks Image
Json Viewer
Bar Viewer
Image Comparison
Depthmap Visualizer
Tensor To Densemap
TENSOR
Crop And Resize
Remap Value Range
Const Tensor
DepthPipeline
IMAGE PROC
Crop
Resize
Brightness
Contrast
Blur


Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera

Provide a visual programming platform for rapidly building ML prototypes

1

Support real-time multimedia user input in-the-wild

2

Provide interactive data augmentation

3

Compare model outputs and
render results directly side-by-side

4

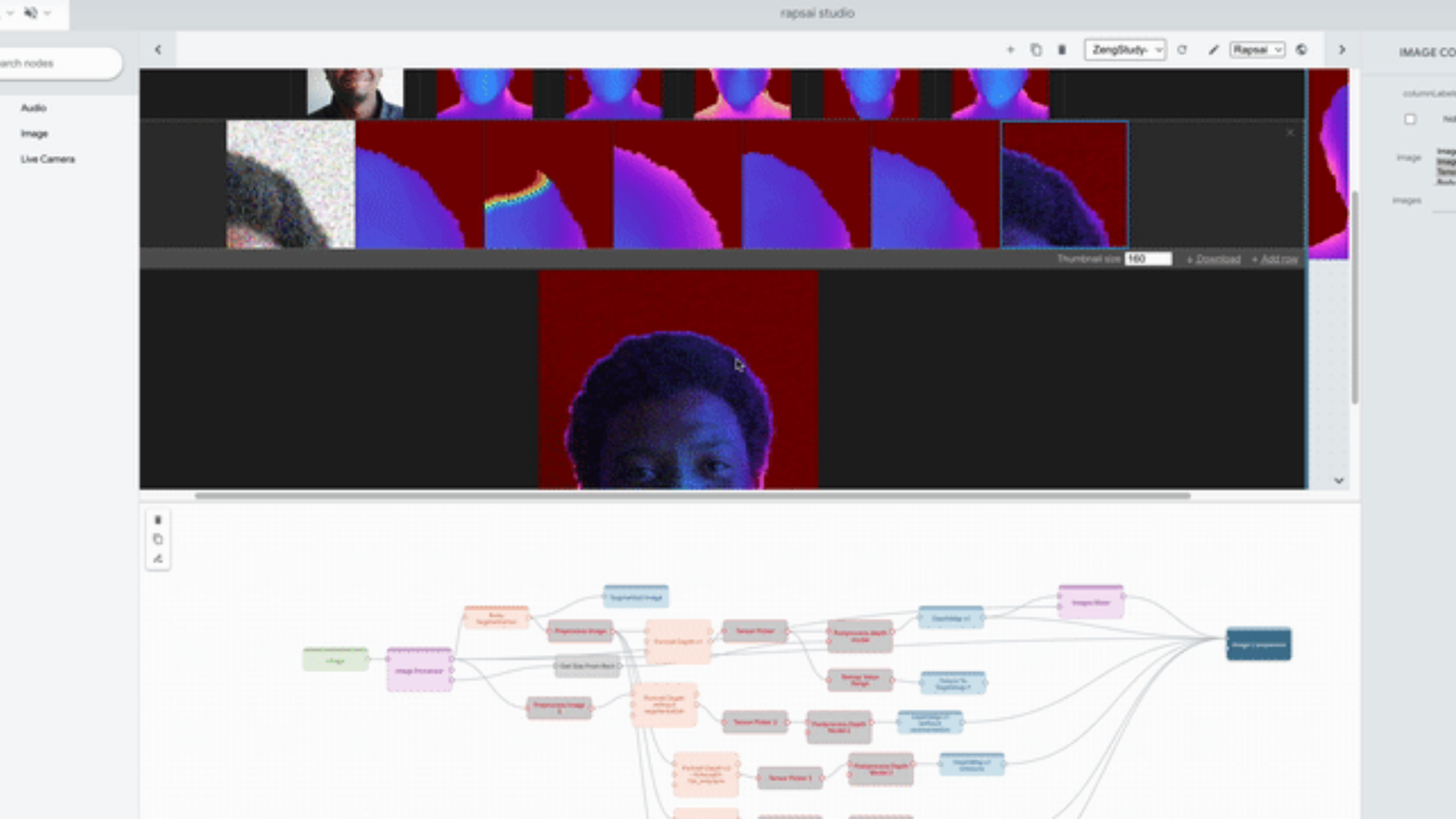
Share visualization with minimum efforts

5

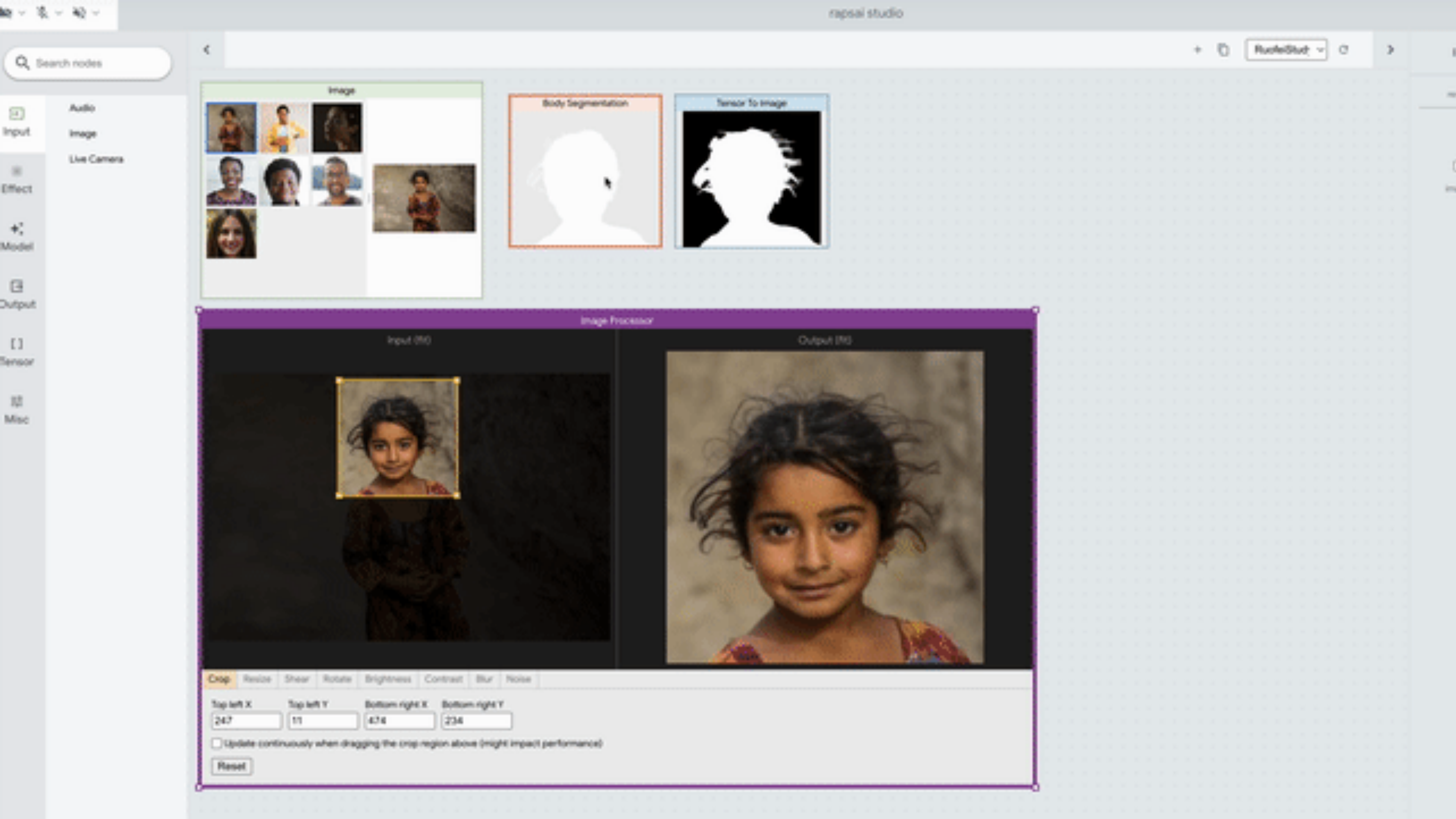
Provide off-the-shelf models and datasets

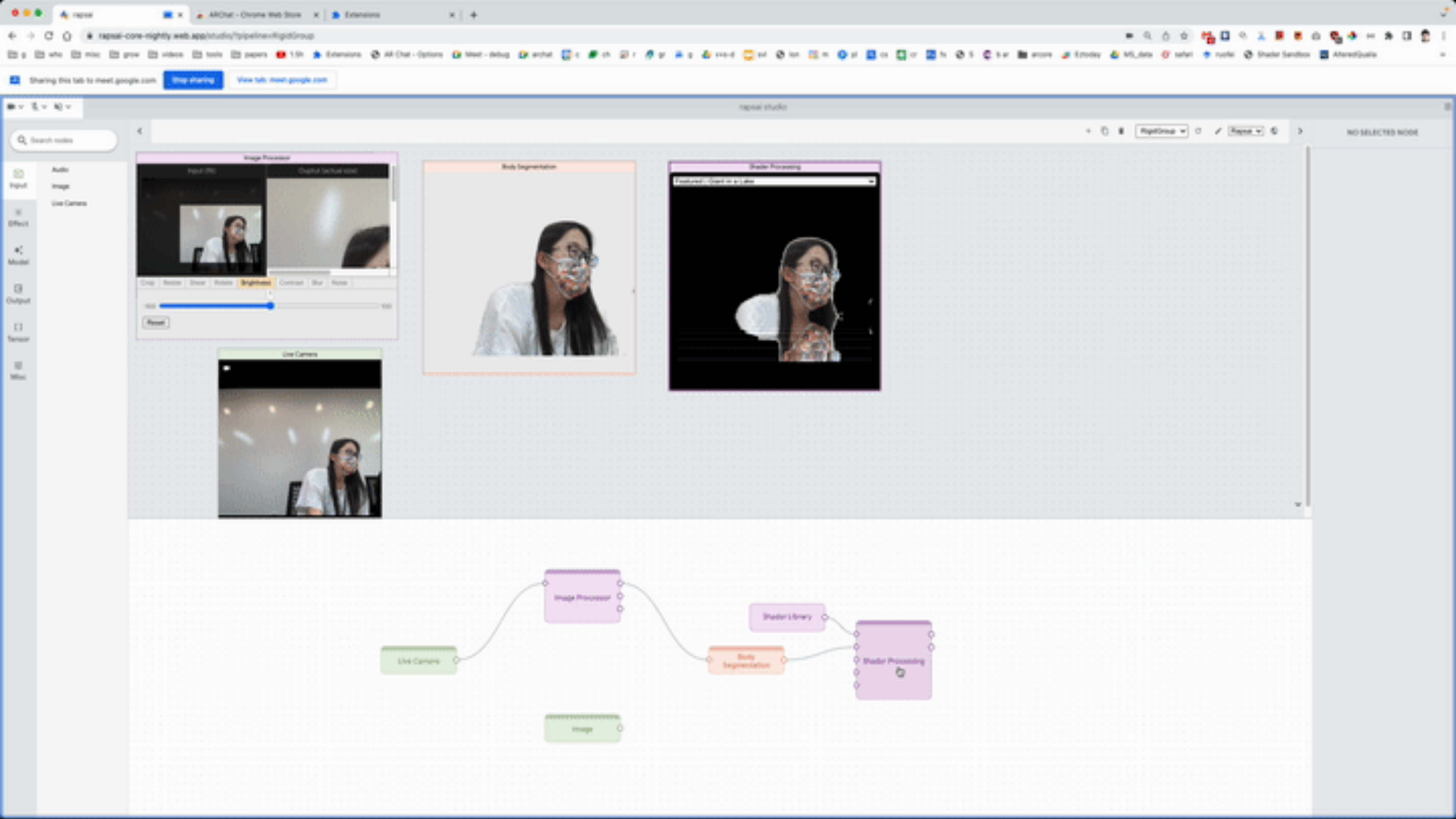
6

There are some times when you don’t want all noise cancellation. People sometimes prefer audio with less noise cancellation because you want some context.

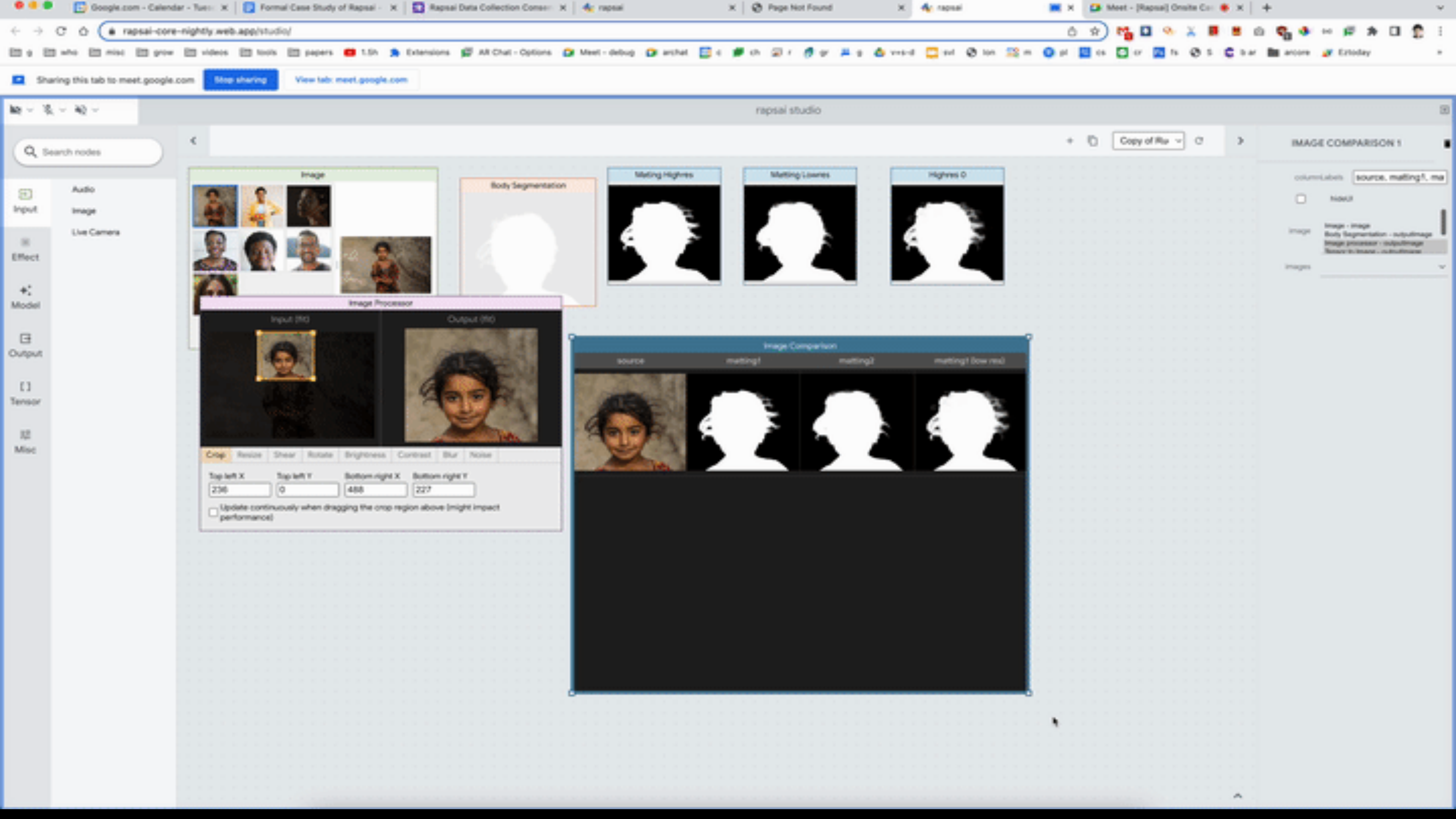


rapsai studio
Audio
Image
Live Camera
Rapsai
Thumbnail size
160
+ Download
+ Add row

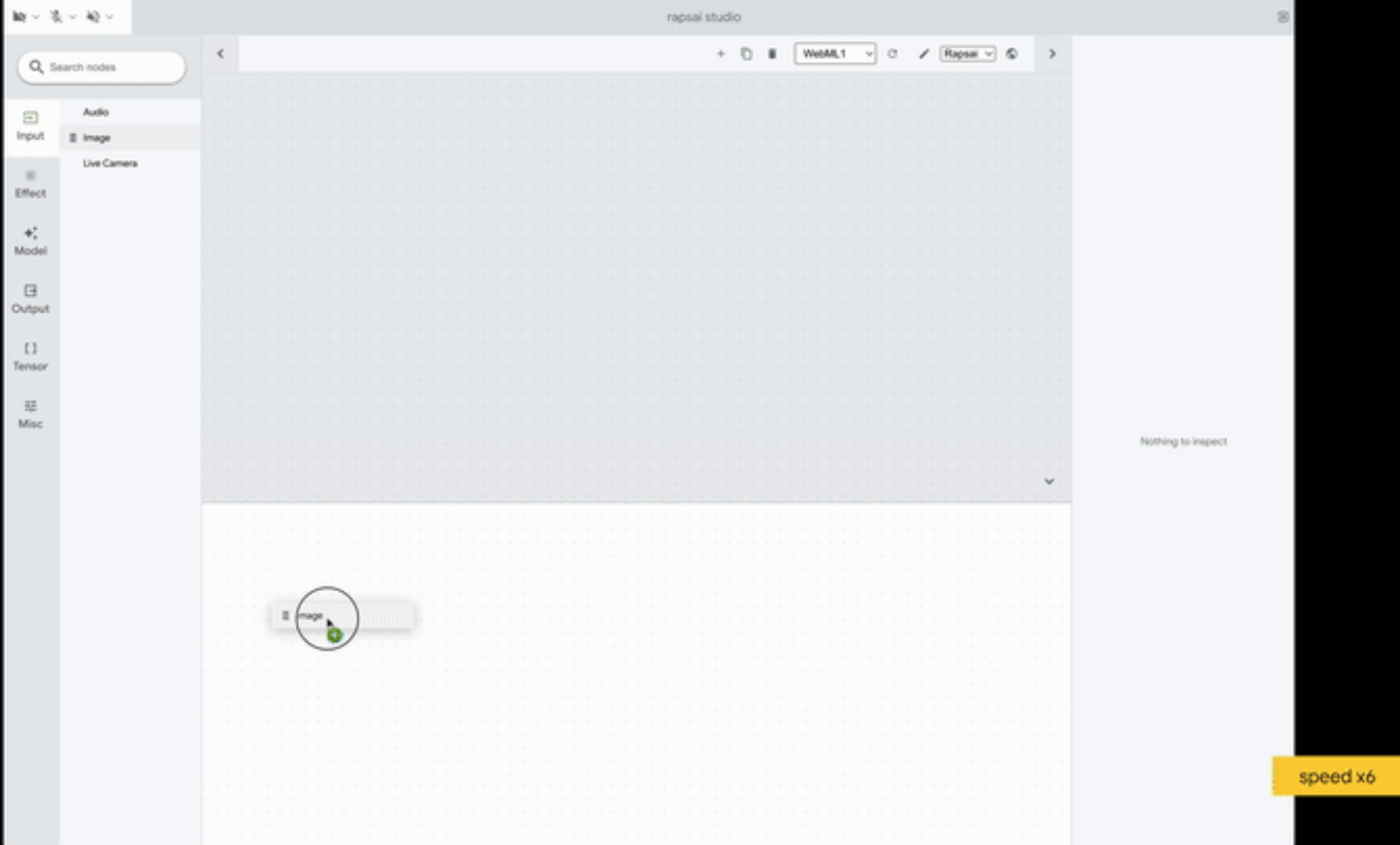

rapsai studio
Search nodes
Input
Audio
Image
Live Camera
Effect
Model
Output
Misc
Image
Body Segmentation
Tensor To Image
Image Processor
Crop
Resize
Shear
Rotate
Brightness
Contrast
Blur
Noise
Top left X
Top left Y
Bottom right X
Bottom right Y
247
11
474
234
Update continuously when dragging the crop region above (might impact performance)
Reset

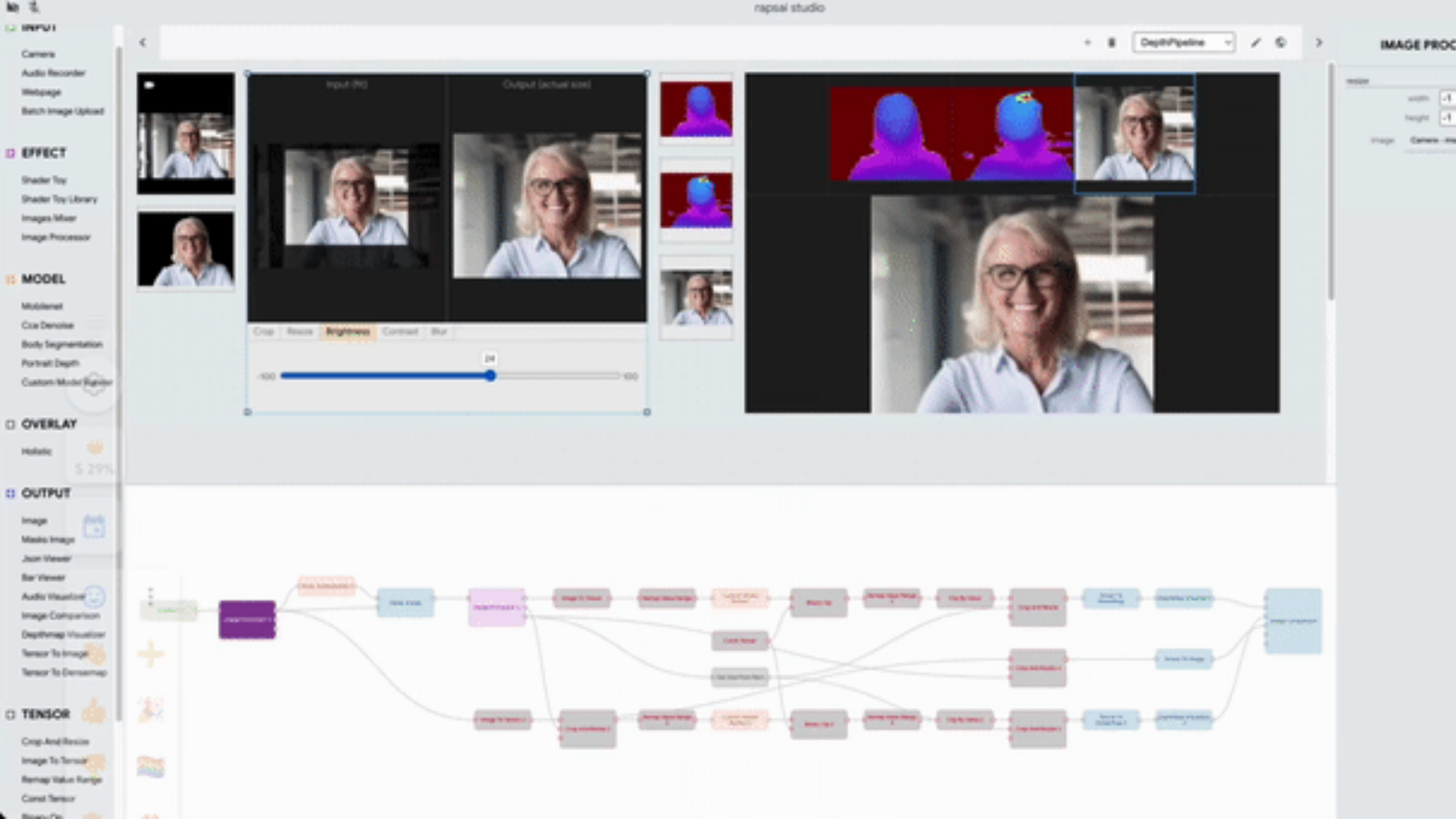


Sharing this tab to meet.google.com
Stop sharing
rapsai studio
NO SELECTED NODE
Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera
Image Processor
Body Segmentation
Shader Processing
Live Camera
Reset
Image Processor
Shader Library
Live Camera
Body Segmentation
Shader Processing
Image

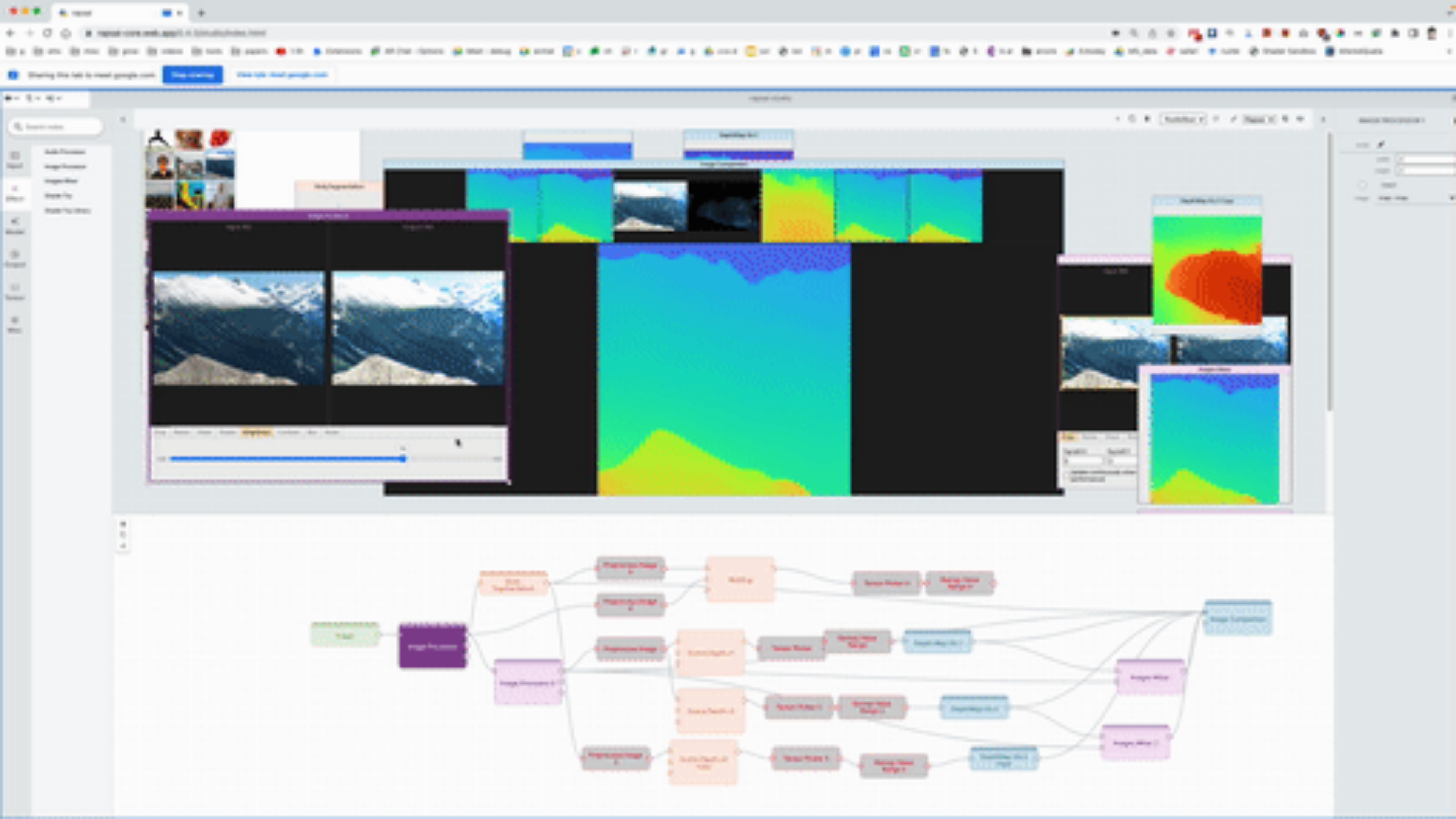


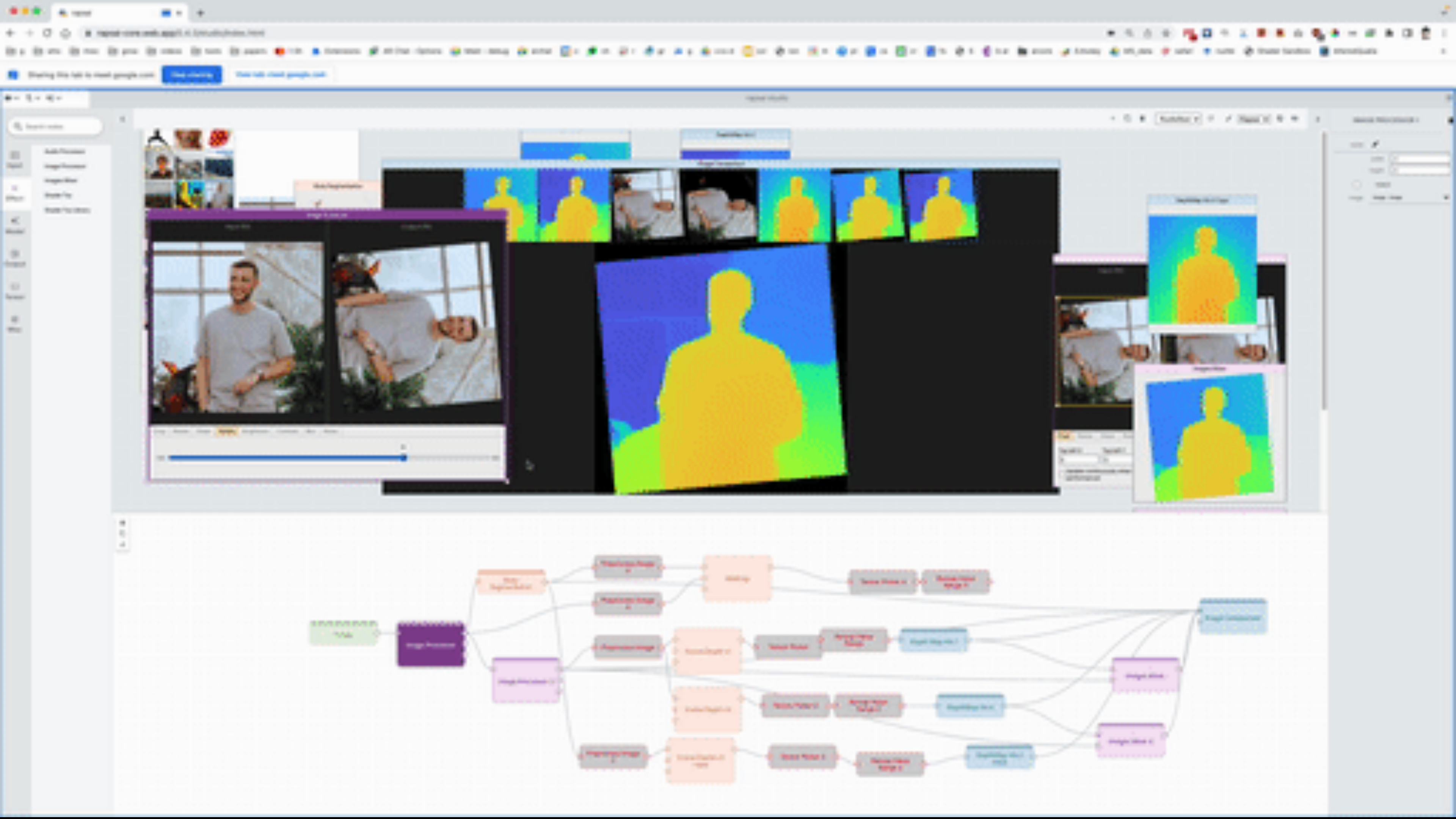
rapsai-core-nightly.web.app/studio/
Sharing this tab to meet.google.com
Stop sharing
View tab: meet.google.com
rapsai studio
Search nodes
Input
Effect
Model
Output
Tensor
Misc
Audio
Image
Live Camera
Copy of Ra
IMAGE COMPARISON 1
Image
Body Segmentation
Matting Highres
Matting Lowres
Highres 0
Image Processor
Input (fit)
Output (fit)
Crop
Resize
Shear
Rotate
Brightness
Contrast
Blur
Noise
Top left X
Top left Y
Bottom right X
Bottom right Y
236
0
488
227
Update continuously when dragging the crop region above (might impact performance)
Image Comparison
source
matting1
matting2
matting1 (low res)

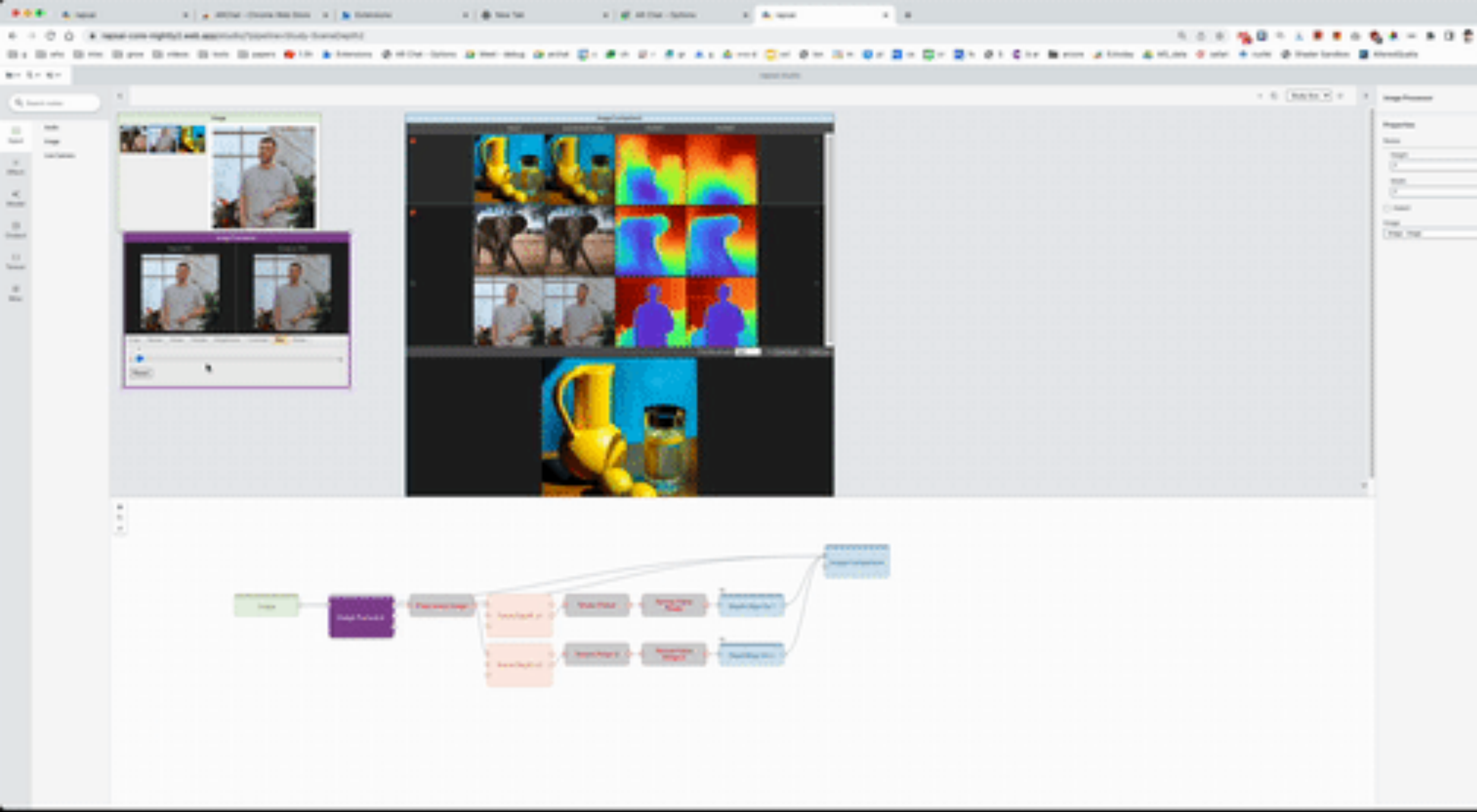

rapsai studio
Search nodes
Audio
Image
Live Camera
Input
Effect
Model
Output
Tensor
Misc
WebML1
Rapsai
Nothing to inspect
Image
speed x6



INPUT
Camera
Audio Recorder
Webpage
Batch Image Upload
EFFECT
Shader Toy
Shader Toy Library
Images Mixer
Image Processor
MODEL
Mobilenet
Body Segmentation
Portrait Depth
OVERLAY
Holistic
OUTPUT
Image
Masks Image
Json Viewer
Bar Viewer
Image Comparison
Depthmap Visualizer
Tensor To Image
Tensor To Densemap
TENSOR
Crop And Resize
Image To Tensor
Remap Value Range
Const Tensor
DepthPipeline
IMAGE PROC
Crop
Resize
Brightness
Contrast
Blur







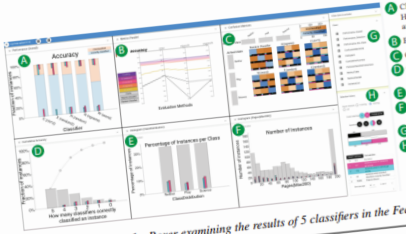
Volume 39 (2020), Number 3

Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Boxer: Interactive Comparison of Classifier Results

Michael Gleicher, Aditya Barve, Xinyi Yu, and Florian Heimerl

University of Wisconsin – Madison



A Classifier Performance view shows a selected metric for all classifiers. Here, the accuracy metric is used and errors for the first two classifiers are clicked as the cyan and magenta selections.
B Parallel Metrics view shows performance for all classifiers for all metrics.
C Confusion Matrix Grid shows performance details for all classifiers.
D Cumulative Accuracy view shows the number of models correctly classifying each instance.
E Histogram view showing the instance distribution over the classes.
F Histogram view showing the instance distribution over the Pages feature.
G View selection palette allows more views to be added to the workspace.
H Selection Controls panel shows the current selections, their relationship, and the selection history. It includes controls for performing set arithmetic.

Figure 1: *Boxer examining the results of 5 classifiers in the Feature Selection use case of Section 5.2.*

Abstract
Machine learning practitioners often compare the results of different classifiers to help select, diagnose and tune models. We present Boxer, a system to enable such comparison. Our system facilitates interactive exploration of the experimental results obtained by applying multiple classifiers to a common set of model inputs. The approach focuses on allowing the user to identify interesting subsets of training and testing instances and comparing performance of the classifiers on these subsets. The system couples standard visual designs with set algebra interactions and comparative elements. This allows the user to compose and coordinate views to specify subsets and assess classifier performance on them. The flexibility of these compositions allow the user to address a wide range of scenarios in developing and assessing classifiers. We demonstrate Boxer in use cases including model selection, tuning, fairness assessment, and data quality diagnosis.

CCS Concepts
• ***Human-centered computing*** → *Visualization; Visual analytics; Information visualization;*

1. Introduction

Machine learning practitioners often perform experiments that compare classification results. Users gather the results of different classifiers or data perturbations on a collection of testing examples. Results are stored and analyzed for tasks such as model selection, hyper-parameter tuning, data quality assessment, fairness testing, and gaining insight about the underlying data. Classifier com- [...] insight about the underlying [...] summary statistics

miss important aspects of classifier performance. For closer examination, practitioners rely on scripting in their standard workflows. The lack of specific tooling makes the process laborious and comparisons challenging, limiting how often experiments are examined in detail.

This paper presents *Boxer* (Figure 1), a system for the detailed examination of classifier comparison experiments. Our approach allows a user to explore a collection of classifier results to identify interesting subsets of the data and compare performance across [...]. Our system enhances standard views with interactions for se- [...] provides a uniform mechanism