# ColourPaletteExtractor

Release 0.5.5

**Tim Churchfield** 

Aug 15, 2021

# Table of Contents

1	colourpaletteextractor  1.1 colourpaletteextractor package	. 1
	Python Module Index	23
	Index	25

# colourpaletteextractor

# 1.1 colourpaletteextractor package

## 1.1.1 Subpackages

## colourpaletteextractor.controller package

Submodules

colourpaletteextractor.controller.controller module

#### class

 $colour palette extractor.controller.controller.Colour Palette Extractor Model. \ colour palette extractor.model.model.Colour Palette Extractor Model, \ view: \ colour palette extractor.view.-main view. Main View )$ 

Bases: PySide2.QtCore.QRunnable

ColourPaletteExtractor Controller.

Used to connect the ColourPaletteExtractor GUI signals with the appropriate slot to be able to manipulate the associated model.

**Parameters** 

- model (ColourPaletteExtractorModel) The main model of ColourPaletteExtractor.
- view (MainView) The main window of ColourPaletteExtractor.

#### current\_tab\_changed(i:int)

Update the current tab index and update the view with the tab's properties.

In most cases,  $i \ge 0$ , however a value of i = -2 or -3 is also valid for performing a 'dummy' tab change to update the current view shown to the user. A value of -1 will lead to the creation of the default tab (the quick start guide).

**Parameters** i (int) – Index of the current tab.

**Raises** ValueError – If the value of i is less than -3.

colourpaletteextractor.controller.worker module

class colourpaletteextractor.controller.worker.Worker ( fn, function\_type: str, tab:
colourpaletteextractor.view.tabview.NewTab, \*args, \*\*kwargs )

Bases: PySide2.QtCore.QRunnable

Worker thread used to generate the colour palette or report for an image.

Inherits from QRunnable to handler worker thread setup, signals and wrap-up.

Adapted from: ref Accessed: 01/08/21

**Parameters** • fn – The function or method to be run as a new thread (generating an image's

colour palette or its colour palette report.

- tab (NewTab) tabview.NewTab object associated with the image to be processed.
- **function\_type** (*str*) The action to be run. This can either be 'colour palette' or 'report'.
- \*args Arguments to pass to the callback function
- \*kwargs Keywords to pass to the callback function

#### Attributes:

**Parameters progress\_callback** – The function callback to run on this worker thread. Supplied args and kwargs will be passed through to the runner.

**Raises** ValueError – If the provided function\_type is invalid.

run ( )

Initialise the runner function with passed args, kwargs.

class colourpaletteextractor.controller.worker.WorkerSignals

Bases: PySide2.QtCore.QObject

Specify the signals available from a running Worker thread.

Adapted from: ref Accessed: 01/08/21 Supported signals are:

#### error

Tuple (exc\_type, value, traceback.format\_exc()).

#### finished

Integer emitted upon finishing.

When generating a colour palette, the value is -2. When generating a report, the value is -3. This is used to reload the tab displaying the image with the correct settings and colour palette.

## progress

NewTab object for which the GUI is to be updated for and the percentage complete for the current task.

The current task is either generating the colour palette for an image or generating the colour palette report for the image.

#### result

Object data returned from processing, anything - NOT IN USE.

staticMetaObject = <PySide2.QtCore.QMetaObject object>

Module contents

#### colourpaletteextractor.examples package

Submodules

 $colour palet te extractor. examples. generate colour palet te example \ module$ 

Contains an example script demonstrating how to generate the colour palette of a sample image. *Module contents* 

## colourpaletteextractor.model package

Subpackages

colourpaletteextractor.model.algorithms package

Submodules

colourpaletteextractor.model.algorithms.cielabcube module

Bases: object

A cube representing a fixed region in the CIELAB colour space.

The cube is used to hold pixels in an image that exist within the cube's region of the CIELAB colour space. The input parameters do not refer to the actual L\*, a\* and b\* values, but depend on the CUBE\_SIZE specified by the colour palette algorithm (in particular, any variant on the Nieves 2020 algorithm).

In the case of the nieves2020. Nieves2020CentredCubes algorithm, the coordinates refer to the centre of the cube. For the nieves2020. Nieves2020OffsetCubes algorithm, the coordinates refer to the corner of the cube closest to the origin.

**Parameters** 

- **1\_star\_coord** (*int*) Perceptual lightness cube coordinate
- a\_star\_coord (int) Green-red cube coordinate
- **b\_star\_coord** (*int*) Blue-yellow cube coordinate

**add\_pixel\_to\_cube** (  $pixel: numpy.array, c\_star: numpy.float64$  )  $\rightarrow$  None Assign a pixel to the cube.

**Parameters** 

- pixel (np.array) The pixel as a [L\*,a\*,b\*] triplet.
- **c\_star** (*np.float64*) The C\* (chroma, relative saturation) value for the pixel.

## property c\_stars: list

The C\* (chroma, relative saturation) values for all of the pixels in the cube.

 $C^{*} = \sqrt{a^{*}}^{2} + \{b^{*}}^{2}$ 

**Returns** (list[np.float64]) – The list of C\* values for all pixels in the cube.

## $calculate\_mean\_colour() \rightarrow None$

Calculate the mean colour of the pixels in the cube.

Nothing is calculated if the number of pixels in the cube is equal to 0.

## property coordinates: numpy.array

The coordinates of the cube ( $[L^*, a^*, b^*]$ ).

In the case of the nieves2020.Nieves2020CentredCubes algorithm, the coordinates refer to the centre of the cube. For the nieves2020.Nieves2020OffsetCubes algorithm, the coordinates refer to the corner of the cube closest to the origin.

**Returns** (np.array) – The cube's coordinates.

 ${\tt get\_c\_star\_percentile\_value} \ ( \ \textit{percentile: float} \ ) \rightarrow {\tt Union[int, numpy.percentile]}$ 

Returns the C\* value for the given percentile based on the pixels in the cube.

**Parameters percentile** (float) – The percentile to calculate the  $C^*$  value for.

**Returns** (Union[int, np.percentile]) – The C\* value for the chosen percentile. If no pixels are found, the return value is 0.

 ${\tt get\_l\_star\_percentile\_value} \ ( \ \textit{percentile: float} \ ) \rightarrow {\tt Union[int, numpy.percentile]}$ 

Returns the L\* value for the given percentile based on the pixels in the cube.

**Parameters percentile** (float) – The percentile to calculate the L\* value for.

**Returns** (Union[int, np.percentile]) – The L\* value for the chosen percentile. If no pixels are found, the return value is 0.

## $increment\_pixel\_count\_after\_reassignment() \rightarrow None$

Increase the number of pixels with this cube's mean colour by one.

## property 1\_stars: numpy.array

The L\* values for all pixels in the cube.

**Returns** (np.array) – Array of L\* values for all pixels in the cube.

## property mean\_colour: numpy.array

The mean colour of the pixels in the cube.

**Returns (np.array)** – The mean colour of the cube as a  $[L^*,a^*,b^*]$  triplet.

## property pixel\_count\_after\_reassignment: int

The number of pixels in the recoloured image with this cube's mean colour.

**Returns (int)** – The number of pixels with the cube's mean colour.

### property pixels: list

The list of pixels ( $[L^*, a^*, b^*]$  triplets) in the cube.

**Returns** (list[np.array]) – The list of pixels in the cube.

## property relevant: bool

The relevancy status of the cube.

**Returns (bool)** – True if the cube is a relevant cube. Otherwise False.

colourpaletteextractor.model.algorithms.cielabcube.get\_relative\_frequencies ( $relevant\_cubes: list, total\_pixels: int$ )  $\rightarrow$  list

Calculate the relative frequency of each colour (relevant colour) in the recoloured image.

## **Parameters**

- relevant\_cubes (list[CielabCube]) List of relevant CielabCube objects.
- total\_pixels (int) The total number of pixels in the image.

**Returns** (list[float]) – The list of relative frequencies for each relevant cube.

colourpaletteextractor.model.algorithms.nieves2020 module

```
{\bf class} \; {\tt colourpaletteextractor.model.algorithms.nieves 2020.} \\ {\bf Nieves 2020} \; ( \; {\it name, url} \; ) \\
```

### Bases:

 $\verb|colour-palette extractor.model.algorithms.palette algorithm.Palette Algorithm|, abc. ABC|$ 

Abstract class representing an algorithm to extract the colour palette from an image.

Based on the algorithm proposed by Nieves et al. (2020); see algorithm for more information.

## $\texttt{CUBE\_SIZE} = 20$

Default delta-E\* length of cube side (units).

## $C_STAR_PERCENTILE = 50$

x-th percentile of C\* (50%) for secondary relevancy requirements.

#### $MIN_L_STAR = 80$

Minimum L\* value for secondary relevancy requirements (units).

## SECONDARY\_THRESHOLD = 0.00375

Minimum secondary pixel count (%) for secondary relevancy requirements.

#### THRESHOLD = 0.03

Minimum threshold colour in each cube (0.03 = 3%) for primary relevancy requirements.

## generate\_colour\_palette ( image: numpy.array ) → tuple

Generate the colour palette and the recoloured image of the provided image.

**Parameters image** (np.array) – The image for which the colour palette is to be generated (in sRGB colour space).

#### Returns

- (np.array) The recoloured image using only the colours in the colour palette.
- (list[np.array]) The list of colours ([R,G,B] triplets) in the image's colour palette.
- (list[float]) The relative frequencies of the colours in the recoloured image.

#### class

colourpaletteextractor.model.algorithms.nieves2020.Nieves2020CentredCubes Bases: colourpaletteextractor.model.algorithms.nieves2020.Nieves2020 Subclass of Nieves2020 with the cube coordinates corresponding to the centre of the cube.

As a result, there is only one cube that touches the origin in the CIELAB colour space (is in fact centred on the

origin).

NAME = 'Nieves, Gomez-Robledo, Chen and Romero (2020) - Cube centred on CIELAB origin' Name of the algorithm.

## URL = 'https://doi.org/10.1364/AO.378659'

Link to more information about the algorithm.

#### class

 $\verb|colour| palette extractor.model.algorithms.nieves 2020. \textbf{Nieves 2020Offset Cubes}|$ 

Bases: colourpaletteextractor.model.algorithms.nieves2020.Nieves2020

Subclass of  ${\tt Nieves2020}$  with the cube coordinates corresponding to the cube's corner closest the origin.

As a result, there are eight cube that touch the origin in the CIELAB colour space.

NAME = 'Nieves, Gomez-Robledo, Chen and Romero (2020) - Cube corners at CIELAB origin' Name of the algorithm.

#### URL = 'https://doi.org/10.1364/AO.378659'

Link to more information about the algorithm.

colourpaletteextractor.model.algorithms.nieves2020.convert\_lab\_2\_rgb ( image: numpy.array )  $\rightarrow$  numpy.array

Convert an image from the CIELAB colour space to the sRGB colour space.

After conversion, the image is scaled to 8-bit per colour channel (24-bit image).

Illuminant = D65 (name of the illuminant) Observer = 2 (aperture angle of observer)

**Parameters image** (np.array) – The image in the CIELAB colour space.

**Returns** (np.array) – The image in the sRGB colour space.

colourpaletteextractor.model.algorithms.nieves2020.convert\_rgb\_2\_lab ( image: numpy.array )  $\rightarrow$  numpy.array

Convert an image from the sRGB colour space to the CIELAB colour space.

The alpha channel of the image (RGBA) is removed if present.

Illuminant = D65 (name of the illuminant) Observer = 2 (aperture angle of observer)

**Parameters image** (np.array) – The image in the sRGB colour space.

**Returns** (np.array) – The image in the CIELAB colour space.

colourpaletteextractor.model.algorithms.nieves2020.get\_c\_stars ( lab: numpy.array )  $\rightarrow numpy.array$ 

Get the matrix of C\* (chroma) values for each pixel in the image.

 $C^{*} = \sqrt{a^{*}}^{2} + \{b^{*}\}^{2}}$ 

**Parameters lab** (np.array) – The image in the CIELAB colour space.

**Returns** (np.array) – Array of C\* values corresponding to each pixel in the image.

colourpaletteextractor.model.algorithms.palettealgorithm module

#### class

colour paletteextractor.model.algorithms.palettealgorithm. Palette Algorithm ( name: str, url: str )

Bases: abc.ABC

Abstract class representing an algorithm used to obtain a colour palette from an image.

**Parameters** 

- name (str) Name of the algorithm
- url (str) Link to a description of the algorithm

## property continue\_thread: bool

Get the execution status of the algorithm.

A value of *false* would indicate that the algorithm should return without generation a colour palette when it next checks its execution status.

Returns bool – The execution status of the algorithm

## abstract generate\_colour\_palette ( image: numpy.array ) → tuple

Generate the colour palette for the given image.

Analyses the given image to obtain its colour palette. Returns the recoloured image using only the colours found in the colour palette, the colour palette of the image and finally the relative frequencies of each of those colours in the recoloured image.

**Parameters image** (np.array) – A 3D array representing an image. It is assumed that the input image is

**Returns** 

- **recoloured\_image** (*np.array*) The recoloured image using only the colours found in the colour palette
- **colour\_palette** (*list*) The list of colours (sRGB 8-bit values) in the colour palette
- relative\_frequencies (*list*) The relative frequencies of each colour in the colour palette in the recoloured image

**Note** It is assumed that the input image has been encoded in the sRGB colour space.

#### property name: str

Get the name of the algorithm.

**Returns (str)** – The name of the algorithm

 $\begin{tabular}{ll} \textbf{set\_progress\_callback} & (&progress\_callback: & PySide2.QtCore.SignalInstance, & tab: \\ colourpaletteextractor.view.tabview.NewTab, image\_data &) \rightarrow \textbf{None} \\ \end{tabular}$ 

Set the signal function called by the algorithm at regular intervals to update the GUI thread.

#### **Parameters**

- progress\_callback (QtCore.SignalInstance) Signal that when emitted, is used to update the GUI.
- tab (NewTab) The tab associated with the image being analysed (see

```
generate_colour_palette().
```

• image\_data (ImageData) - ImageData object that holds the image being analysed.

## property url: str

Get the link to the description of the algorithm.

**Returns (str)** – The link to the description of the algorithm

colourpaletteextractor.model.algorithms.palettealgorithm.get\_implemented\_algorithms
( )

Recursively finds all subclasses of the PaletteAlgorithm class.

Like Python's \_\_class\_.\_subclasses\_\_(), but recursive. Returns a list containing all subclasses of PaletteAlgorithm.

Adapted from: ref

Accessed: 15/07/2021

#### **Returns:**

[object]: List of all non-abstract subclasses of PaletteAlgorithm

#### Module contents

#### Submodules

colourpaletteextractor.model.generatereport module

class colourpaletteextractor.model.generatereport.ColourPaletteReport (
image\_data: colourpaletteextractor.model.imagedata.ImageData )

Bases: fpdf.fpdf.FPDF, fpdf.html.HTMLMixin

A modified FPDF object to fit the requirements for generating a PDF colour palette report.

**Parameters image\_data** (ImageData) – The ImageData object holding the image's data (the original image, the recoloured image, and the colour palette).

#### A4\_HEIGHT = 297

The height of an A4 sheet of paper (mm).

## A4\_WIDTH = 210

The width of an A4 sheet of paper (mm).

## IMAGE\_START\_POSITION = 30

The standard left indentation when placing an image in the PDF report (mm).

## IMAGE WIDTH = 150

The standard width of images in the PDF report (mm).

#### MARGIN = 10

The size of the margins to be used in the PDF report (mm).

#### MAX\_IMAGE\_HEIGHT = 257

The standard maximum height of images in the report (mm).

#### footer $() \rightarrow None$

Set the footer used in the PDF report.

## $header() \rightarrow None$

Set the header used in the PDF report.

class colourpaletteextractor.model.generatereport.ReportGenerator ( tab: colourpaletteextractor.view.tabview.NewTab, image\_data: colourpaletteextractor.model.imagedata.ImageData, settings: PySide2.QtCore.QSettings, progress\_callback: PySide2.QtCore.SignalInstance)

Bases: object

Class used to create, populate a ColourPaletteReport object and save the resulting PDF to disk.

#### **Parameters**

- tab (NewTab) The tab associated with the image to be analysed.
- image\_data (ImageData) The ImageData object holding the image's data (the original image, the recoloured image, and the colour palette).
- **settings** (*QSettings*) The settings for the ColourPaletteExtraction application.
- progress\_callback (QtCore.SignalInstance) Signal that when emitted, is used to update the GUI.

Create a ColourPaletteReport object representing the PDF colour palette report.

Returns (Union[ColourPaletteReport, None]) - None if the ColourPaletteReport object was not properly generated, otherwise returns the populated ColourPaletteReport object.

save\_report ( pdf: colourpaletteextractor.model.generatereport.ColourPaletteReport ) → None
save the ColourPaletteReport object representing the PDF colour palette report to disk.

Parameters pdf (ColourPaletteReport) - The ColourPaletteReport object to be saved as a PDF to disk..

colourpaletteextractor.model.generatereport.generate\_report ( tab: colourpaletteextractor.view.tabview.NewTab,  $image\_data:$  colourpaletteextractor.model.imagedata.ImageData, settings: PySide2.QtCore.QSettings,  $progress\_callback: PySide2.QtCore.SignalInstance$ )  $\rightarrow$  None Generate a colour palette report for an image.

#### **Parameters**

- tab (NewTab) The tab associated with the image to be analysed.
- image\_data (ImageData) The ImageData object holding the image's data (the original image, the recoloured image, and the colour palette).
- **settings** (*QSettings*) The settings for the ColourPaletteExtraction application.
- progress\_callback (QtCore.SignalInstance) Signal that when emitted, is used to update the GUI.

**Raises** ValueError – If the provided ImageData object does not have a recoloured image or has no colours in its colour palette.

colourpaletteextractor.model.imagedata module

class colourpaletteextractor.model.imagedata.ImageData (file\_name\_and\_path: str)
 Bases: object

Object to hold the data associated with an image to be analysed.

Stores the original image, its colour palette, the recoloured image, the relative frequency of each colour in the recoloured image, the algorithm used to generate the colour palette and the execution status of the thread used to generate the colour palette.

**Parameters file\_name\_and\_path** (str) – Path to the image to be added.

**Raises** ValueError – If the file\_name\_and\_path argument is None.

## property algorithm\_used: type

The algorithm used to generate the image's colour palette.

**Returns (type[palettealgorithm.PaletteAlgorithm])** – The class name of the colour palette extraction algorithm.

#### property colour\_palette: list

The list of colours in the image's colour palette.

**Returns** (list[np.array]) – The list of colours ([R,G,B] triplets) in the colour palette.

## property colour\_palette\_relative\_frequency: list

The relative frequencies of each colour in the colour palette in the recoloured image.

The order of the relative frequencies matches the order of the colours in the colour palette.

**Returns** (list[float]) – The list of relative frequencies of the colour palette.

## property continue\_thread: bool

Specify if the thread for generating the colour palette or the report should be cancelled.

Returns (bool) - True if the thread should be continued. Otherwise False.

## property extension: str

The file extension of the original image.

**Returns (str)** – The file extension of the original image.

## property file\_name\_and\_path: str

The file path to the original image.

**Returns (str)** – File path to the original image.

## static get\_image\_as\_q\_image ( image: numpy.array ) → PySide2.QtGui.QImage

Convert a Numpy array representation of an image to a QImage.

**Parameters image** (np.array) – An image represented by a Numpy array.

**Returns** (QImage) – The image converted to a QImage.

Raises ValueError – If the provided image is not a greyscale, rGB or RGBA image (1,

3, or 4 colour channels).

## property image: numpy.array

The original image, represented as a 2 or 3-D Numpy array.

**Returns** (np.array) – The original image as a Numpy array.

## property name: str

The name of the image, without its file extension.

**Returns** (str) – The image file name, without its extension.

## property recoloured\_image

The recoloured image, represented as a 3-D Numpy array.

**Returns** (**np.array**) – The recoloured image as a Numpy array.

## sort\_colour\_palette ( reverse: bool = True ) → None

Sort the colour palette by their relative frequencies in the recoloured image.

**Parameters reverse** (bool) – If True, the colour palette is sorted from largest relative frequency to the smallest. If False, the order is smallest to largest. The default is True.

## colourpaletteextractor.model.model module

#### class colourpaletteextractor.model.model.ColourPaletteExtractorModel

Bases: object

ColourPaletteExtractor Model.

Used as the model component of the ColourPaletteExtractor application.

#### DEFAULT\_ALGORITHM

alias of

colourpaletteextractor.model.algorithms.nieves2020.Nieves2020CentredCubes

## **DEFAULT HEIGHT:** int = 894

Default height of the ColourPaletteExtraction application.

Size chosen to show the Quick Start Guide image without the need of scrollbars.

#### DEFAULT\_USER\_DIRECTORY: str = '/Users/tim/Documents/ColourPaletteExtractor/Output'

The default user output directory for colour palette reports.

## DEFAULT\_USE\_USER\_DIRECTORY: bool = False

Specify by default whether a user's output directory should be used for saving the colour palette report to.

#### DEFAULT\_WIDTH: int = 1523

Default width of the ColourPaletteExtraction application.

Size chosen to show the Quick Start Guide image without the need of scrollbars.

# SUPPORTED\_IMAGE\_TYPES: set = {'jpeg', 'jpg', 'png'}

The set of supported image extensions.

#### property active\_thread\_counter: int

The number of active threads still running as part of a batch operation.

**Returns (int)** – The number of active threads still running.

## add\_image ( $file_name_and_path: str$ ) $\rightarrow$ tuple

Given the path to an image, create a new ImageData object and return it and its ID key.

**Parameters file\_name\_and\_path** (str) – Path to the image.

Returns

- (str) The dictionary key ('Tab\_xx') for the new ImageData object in the image\_data\_id\_dictionary.
- (ImageData) The new ImageData object for holding information about the image (e.g., the colour palette, the recoloured image etc.)

Raises KeyError – If the generated dictionary key already exists in the model's dictionary of ImageData objects (image\_data\_id\_dictionary).

## **change\_output\_directory** ( *use\_user\_dir: bool, new\_user\_directory: str* ) → None

Change the output directory for colour palette reports in the ColourPaletteExtractor.ini settings file.

**Parameters** 

- **use\_user\_dir** (*bool*) True if the user-selected output directory is to be used. If False, use default temporary output directory.
- **new\_user\_directory** (*str*) The path to the new user-selected output directory

## $close\_temporary\_directory() \rightarrow None$

Delete the temporary output directory associated with the instance of the application.

 $generate\_palette$  (  $image\_data\_id: str$ ,  $tab: Optional[colourpaletteextractor.view.tabview.NewTab] = None, progress\_callback: Optional[PySide2.QtCore.SignalInstance] = None, algorithm: Optional[type] = None) <math>\rightarrow$  None

Generate the colour palette for the image in the ImageData object with the given image\_data\_id ID.

The recoloured image, colour palette and relative frequencies of each colour are added to the ImageData object with the image\_data\_id dictionary key.

#### **Parameters**

- image\_data\_id (str) The dictionary key/ID ('Tab\_xx') for the ImageData object in the image\_data\_id\_dictionary for which the colour palette of its associated image is to be generated for.
- **tab** (NewTab) The NewTab linked to the image that is to have its colour palette generated.
- progress\_callback (QtCore.SignalInstance) Signal that when emitted, is used to update the GUI.
- algorithm (type[PaletteAlgorithm]) The algorithm class to be used to generate the colur palette.

 $\label{eq:generate_report} \textbf{(}\textit{tab: colourpaletteextractor.view.tabview.NewTab, progress\_callback: } PySide2.Qt-Core.SignalInstance\textbf{)} \rightarrow \textbf{None}$ 

Generate the colour palette report for the image linked to the given NewTab.

## **Parameters**

- tab (NewTab) The NewTab linked to the image that is to have its colour palette report generated.
- progress\_callback (QtCore.SignalInstance) Signal that when emitted, is used to update the GUI.

get\_image\_data ( image\_data\_id: str ) → colourpaletteextractor.model.imagedata.ImageData

Returns the ImageData object with the given ID/key in the image\_data\_id\_dictionary.

Parameters image\_data\_id (str) - The dictionary key/ID ('Tab\_xx') for the ImageData object in the image\_data\_id\_dictionary that should be returned.

**Returns** (ImageData) – ImageData object with the given ID/key.

## property image\_data\_id\_dictionary: dict

The dictionary storing the ImageData objects for the images currently open.

**Returns (dict)** – dictionary storing the ImageData objects for the images currently open.

## static read\_view\_settings() → tuple

Get the size and shape of the main window of the GUI.

## Returns

- **(Optional[QSize])** The size of the main window. None if the appropriate setting cannot be found.
- (Optional[QPoint]) The position of the main window. None if the appropriate setting cannot be found.

## remove\_image\_data ( image\_data\_id: str ) → None

Remove ImageData object from the dictionary of images (image\_data\_id\_dictionary) by its key.

Parameters image\_data\_id (str) - The dictionary key ('Tab\_xx') for the ImageData object in the image\_data\_id\_dictionary that should be removed.

```
set_algorithm ( algorithm\_class: type = < class 'colourpaletteextractor.model.algorithm-s.nieves2020.Nieves2020CentredCubes'>) <math>\rightarrow None
```

Set the algorithm used to generate the colour palette of an image.

If no algorithm\_class\_name is provided, the DEFAULT\_ALGORITHM is used.

Parameters algorithm\_class (type[PaletteAlgorithm]) - The algorithm class.

```
write_default_settings () \rightarrow None
```

Write the default settings to the ColourPaletteExtractor.ini settings file.

 $\textbf{static write\_view\_settings} \ ( \ \textit{size: PySide2.QtCore.QSize, position: PySide2.QtCore.QPoint } \ ) \rightarrow \\ None$ 

Write the main window's size and shape to the settings file.

**Parameters** 

- size (QSize) The size of the GUI.
- **position** (*QPoint*) The position of the GUI.

```
colourpaletteextractor.model.model.generate_colour_palette_from_image path\_to\_file: str, algorithm: Optional[type] = None) 	o tuple
```

Generate the colour palette for the given images using the specified colour palette extraction algorithm.

An example algorithm would be nieves2020. Nieves2020 Centred Cubes

**Parameters** 

- path\_to\_file (str) Path to the image to be analysed.
- **algorithm** (*type*[*PaletteAlgorithm*]) The Python class of the the colour palette extraction algorithm.

Returns

- (np.ndarray) THe recoloured image using just the colours in the colour palette.
- (list[np.ndarray]) The list of colours ([R,G,B] triplets) in the colour palette.
- (list[float]) The relative frequencies of the colours in the colour palette in the recoloured image.

```
colour
paletteextractor.model.model.get_settings ( ) \rightarrow PySide2.QtCore.QSettings
Get the settings file for the ColourPaletteExtraction application.
```

**Returns (QSettings)** – The settings for the ColourPaletteExtraction application.

Module contents

## colourpaletteextractor.tests package

Subpackages

colourpaletteextractor.tests.helpers package

Submodules

colourpaletteextractor.tests.helpers.helperfunctions module

colourpaletteextractor.tests.helpers.helperfunctions.get\_image ( path\_to\_image:
str )

Returns the image found at the given path.

**Parameters path\_to\_image** (*str*) – Path to the image to be imported.

**Returns** (np.array) – Image represented as a 3D array

Module contents

Submodules

```
colourpaletteextractor.tests.nieves2020_test module
colourpaletteextractor.tests.nieves2020_test.test_closest_relevant_colour_used_to_recol
colourpaletteextractor.tests.nieves2020_test.test_cube_colour_must_occur_more_than_thre
( )
colourpaletteextractor.tests.nieves2020_test.test_cube_colour_must_occur_more_than_thre
colourpaletteextractor.tests.nieves2020_test.test_low_a_b_colour_does_not_meet_secondar
colourpaletteextractor.tests.nieves2020_test.test_low_a_b_colour_does_not_meet_secondar
( )
colourpaletteextractor.tests.nieves2020 test.test nieves2020 centred cubes constructor
colourpaletteextractor.tests.nieves2020 test.test nieves2020 offset cubes constructor
( )
colourpaletteextractor.tests.nieves2020_test.test_primary_requirements_1()
colourpaletteextractor.tests.nieves2020_test.test_primary_requirements_2()
colourpaletteextractor.tests.nieves2020_test.test_primary_requirements_3()
colourpaletteextractor.tests.nieves2020_test.test_recoloured_image_of_same_size_1
colourpaletteextractor.tests.nieves2020_test.test_recoloured_image_two_colours_1
colourpaletteextractor.tests.nieves2020_test.test_recoloured_image_two_colours_2
colourpaletteextractor.tests.nieves2020_test.test_two_colours_in_same_cube_can_meet_sec
( )
Module contents
colourpaletteextractor.view package
Submodules
colourpaletteextractor.view.mainview module
            colourpaletteextractor.view.mainview.MainView
Optional[PySide2.QtCore.QSize] = None, position: Optional[PySide2.QtCore.QPoint] = None, parent=None)
   Bases: PySide2.QtWidgets.QMainWindow
   The main window of the ColourPaletteExtractor application.
   Parameters
                • size (QSize) – Size of the main window. Default is None.
                • position (QPoint) – Position of the main window. Default is None.
                • parent – Parent object of the MainWindow. Defaults to None.
   tabs
       tabbed widget for displaying and managing imported images.
```

Type QTabWidget

#### colour\_palette\_dock

Type tabview.ColourPaletteDock

## \_close\_request\_action

Action for closing the application

Type QAction

#### open\_action

Action for opening a new image

Type QAction

#### generate\_report\_action

Action for generating a report for an image

Type QAction

## generate\_all\_report\_action

Action for generating a report for all images with a colour palette

Type QAction

## generate\_palette\_action

Action for generating the colour palette for an image

Type QAction

## generate\_all\_palette\_action

Action for generating the colour palette for all images

Type QAction

#### stop action

Action for stopping the report or colour palette being generated for an image

Type QAction

## preferences\_menu\_action

Action for opening the preferences menu

Type QAction

## show\_help\_action

Action for showing the quick start guide

Type QAction

## toggle\_recoloured\_image\_action

Action for toggling between the original and the recoloured image

Type QAction

## zoom\_in\_action

Action for zooming into an image

Type QAction

## zoom\_out\_action

Action for zooming out of an image

Type QAction

### about\_menu\_action

Action for showing the about information widget

## Type QAction

#### show\_palette\_dock\_action

Action for showing the colour palette dock

**Type** QAction

#### show\_toolbar\_action

Action for showing the toolbar

Type QAction

#### tools

(QToolBar): Toolbar for holding QToolButtons used in the GUI

#### status

Status bar for holding hints, the progress bar and the current version of the application

Type otherviews.StatusBar

## RESOURCES\_DIR = 'resources'

The name of the directory containing the icons and images used for the GUI.

Type str

## app\_icon = 'app\_icon'

The name of the file used as the application's icon.

Type str

## **closeEvent** (*event: PySide2.QtGui.QCloseEvent* ) → None

Intercept GUI close event to check if the user wishes to close the GUI.

Parameters event (QtGui.QCloseEvent) - Close event

## **close\_current\_tab** ( *tab\_index: int* ) → int

Close the tab with the given index.

Parameters tab\_index (int) - The index of the tab to close

**Returns** (int) – The index of the tab that is now visible after closing the selected tab

## create\_new\_tab ( image\_id, image\_data ) → None

Create a new image tab for the main window.

#### **Parameters**

- **image\_id** (str) ID of the image to be used for the new tab (e.g., 'Tab\_1')
- image\_data (model.imagedata.ImageData) Object containing tab and image properties and state

## default\_new\_tab\_image = 'images:how-to-dark-mode.png'

The name of the file used as the default new tab (the quick start guide).

Type str

# resources\_path = '/Users/tim/OneDrive - University of St Andrews/University/MScProject/-ColourPaletteExtractor/colourpaletteextractor/view/resources'

The path to the resources used for the GUI.

This will vary depending on whether the code has been compiled into an application or is been run from the command line.

Type str

```
show_file_dialog_box ( supported_file_types: set ) → tuple
```

Show the dialog box for importing images.

Parameters supported\_file\_types (set[str]) - The supported file types (e.g., '.png')

**Returns** 

- **list** (*str*) The list of the absolute paths to the images to be loaded into the application
- str The filter used when selecting the images to import

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

colourpaletteextractor.view.otherviews module

```
class colourpaletteextractor.view.otherviews.AboutBox ( parent=None )
```

Bases: PySide2.QtWidgets.QMessageBox

Message box to show the basic information about the application.

**Parameters parent** – The parent object of the AboutBox. The default is None.

staticMetaObject = <PySide2.QtCore.QMetaObject object>

## class colourpaletteextractor.view.otherviews.BatchGenerationProgressWidget

Bases: PySide2.QtWidgets.QDialog

Custom dialog box shown when multiple colour palette or reports are being generated.

Shows the number of threads to be run and the number of threads completed. Is also has a simple animation attached to it so the user knows that the application has not frozen and is still processing their images.

#### label

Label used to show the number of threads to be run and the number completed.

Type QLabel

## cancel batch button

The button used to notify the controller object that the user wishes to cancel the current batch processing.

Type QPushButton

#### $set\_cancel\_text() \rightarrow None$

Set the text shown to cancelling to let the user know that any incomplete threads are to be cancelled.

**show\_widget** ( *total\_count*: *int*, *batch\_type*: *str* ) → None

Reset and show the widget.

**Parameters** 

- total\_count (int) The total number of threads to be processed.
- **batch\_type** (*str*) The text clarifying what task is being carried out as a batch process.

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

```
update\_progress() \rightarrow None
```

Update the batch progress bar by increasing the number of completed threads by one.

class colourpaletteextractor.view.otherviews.ElidedLabel ( text='', width=40,
parent=None)

Bases: PySide2.QtWidgets.QLabel

Status bar message label that will become elided if there is not enough space to display the entire message.

Adapted from: ref1 and ref2

Accessed: 18/07/2021

#### **Args:**

text (str): The text to be shown in the label. The default is an empty string width (int): The minimum width of the label. The default is 40.

parent: The parent object of the ElidedLabel. The default is None.

## elided\_text() → str

Get the elided text shown by the label.

Returns (str) - The elided text

## **paintEvent** ( event: PySide2.QtCore.QEvent.Type.Paint ) → None

Update the text shown by the label on receiving a paint event.

Parameters event (QEvent. Type. Paint) - A paint event

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

class colourpaletteextractor.view.otherviews.ErrorBox ( box\_type: Optional[str] = None,
parent=None )

Bases: PySide2.QtWidgets.QMessageBox

Message box to show warnings and errors.

## **Parameters**

- **box\_type** (*str*) The error box type. Used to customise the icon and main text show.
- parent Parent object of the ErrorBox. Defaults to None.

#### header

The heading of the ErrorBox.

Type str

```
append_title ( error: Exception ) → None
```

Append the title with additional information from an exception.

**Parameters error** (*Exception*) – Exception whose error summary message is appended to the title text.

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

```
class colourpaletteextractor.view.otherviews.PreferencesWidget (parent=None)
```

Bases: PySide2.QtWidgets.QDialog

The dialog box for setting a user's preferences.

Currently, the user can change the algorithm used to generate the colour palette, as well as the output directory for any reports that are generated.

**Parameters parent** – The parent object of the PreferencesWidget. The default is None.

#### browse button

Button used to open the operating system's file explorer to select a valid output directory.

Type QPushButton

#### user\_path\_selector

Text window used to show the user's currently selected output directory.

Type QLineEdit

#### default\_path\_button

Button used to select the default output directory.

```
Type QRadioButton
```

## user\_path\_button

Button used to select the user's output directory.

Type QRadioButton

#### output\_tab

The output directory settings tab of the preferences dialog box.

Type QWidget

#### algorithm\_tab

The algorithm settings tab of the preferences dialog box.

Type QWidget

## $\mathtt{get\_algorithms\_and\_buttons}\ (\ ) \to \mathtt{tuple}$

Get the list of algorithm classes and their associated buttons.

Returns

- (list[palettealgorithm.PaletteAlgorithm]) List of algorithm classes.
- (list[QRadioButton]]) List of buttons associated with the algorithm classes.

```
show_output_directory_dialog_box ( current_path: str )
```

Show the dialog box for selecting output directory for reports.

**Parameters current\_path** (*str*) – The path to open the system's file explorer to.

**Returns** (str) – Path to the new output directory.

```
show\_preferences() \rightarrow None
```

Show the preferences widget.

```
staticMetaObject = <PySide2.QtCore.QMetaObject object>
```

```
update_preferences () \rightarrow None
```

Update the preferences dialog box with the correct settings.

```
class colourpaletteextractor.view.otherviews.StatusBar ( parent=None )
```

Bases: PySide2.QtWidgets.QStatusBar

The status bar at the bottom of the main window.

This holds the current shortcut tip for the given tab, as well as the progress bar for showing the current progress towards generating a report or the image's colour palette.

**Parameters parent** – Parent object of the StatusBar. Defaults to None.

#### status label

Primary status label.

Type ElidedLabel

#### \_progress\_bar

Progress bar used to track the progress of generating a colour palette or a report.

Type QProgressBar

## \_max\_progress

Maximum value for the progress bar.

Type int

## \_min\_progress

Minimum value for the progress bar.

Type int

#### **set\_status\_bar** ( *state: int* ) → None

Set the state of the status bar elements.

Depending on the state, the primary status label will change to reflect what the application is currently processing.

**Parameters state** (*int*) – The new state of the status bar.

**Raises** ValueError – If state is not a valid state.

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

```
update\_progress\_bar(n:float) \rightarrow None
```

Update the current level of progress for the status bar.

**Parameters n** (float) – New level of progress for the progress bar.

**Raises** ValueError – If the new progress value exceeds the predefined limits of the progress bar.

colourpaletteextractor.view.tabview module

```
class colourpaletteextractor.view.tabview.ColourBox ( parent=None )
```

Bases: PySide2.QtWidgets.QLabel

Modified QLabel to hold an individual colour in the colour palette.

**Parameters parent** – The parent object of the ColourBox. The default is None.

```
enterEvent ( event: PySide2.QtCore.QEvent ) → None
```

Intercept an enter event.

In the future, this could be used to trigger the highlighting regions of the image that use this colour in the recoloured image.

**Parameters event** (*QEvent*) – Enter event.

```
leaveEvent ( event: PySide2.QtCore.QEvent )
```

Intercept a leave event.

In the future, this could be used to cancel the highlighting of regions of the image that use this colour in the recoloured image.

Parameters event (QEvent) – Leave event.

#### staticMetaObject = <PySide2.QtCore.QMetaObject object>

```
class colourpaletteextractor.view.tabview.ColourPaletteDock ( parent=None )
```

Bases: PySide2.QtWidgets.QDockWidget

A modified QDockWidget to hold small images of each colour in an image's colour palette.

Parameters parent - Parent object of the ColourPaletteDock. Defaults to None.

 $add\_colour\_palette$  (  $colour\_palette$ : list,  $image\_id$ : str,  $relative\_frequencies$ : Optional[list] = None )  $\rightarrow$  None

Clear the colour palette dock and add a new image's colour palette to the dock.

#### **Parameters**

- **colour\_palette** (*list[np.array]*) List of colours in the colour palette.
- $image_id(str)$  The ID ('Tab\_xx') associated with a tab and image.
- **relative\_frequencies** (*list[float]*) The relative frequencies of each colour in the colour palette in the recoloured image.

```
remove\_colour\_palette() \rightarrow None
```

Remove all of the ColourBox labels from the colour palette dock.

Adapted from: ref Accessed: 27/07/2021

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

class colourpaletteextractor.view.tabview.ImageDisplay ( image\_data:
 colourpaletteextractor.model.imageData, parent=None )

Bases: PySide2.QtWidgets.QLabel

A modified QLabel to display and manipulate the current image.

#### Parameters

- image\_data (imagedata.ImageData) The ImageData object that hold the information associated with an image.
- parent Parent object of the ImageDisplay. Defaults to None.

```
event (event: PySide2.QtCore.QEvent ) → bool
```

Intercept the QLabel's event if it is a gesture to allow for zooming into and out of the current image.

Also calls the super class' event handler at the end.

**Parameters event** (*QEvent*) – An event.

**Returns (bool)** – The result from the super class' event handler.

 $image\_zoom$  (  $mouse\_pos: PySide2.QtCore.QPoint, value: float ) <math>\rightarrow$  None

Zoom into or out of an image at the mouse pointer's current location.

**Parameters** 

- mouse\_pos (QtCore.QPoint) Current position of the mouse cursor.
- **value** (*float*) The degree of magnification of the image.

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

```
update_image ( image: numpy.array ) → None
```

Update the image shown by the ImageDisplay.

**Parameters image** (np.array) – Numpy array representing an image.

```
zoom factor = 1.25
```

The zoom-in factor used when the user zoom's into the image via the zoom-in button.

```
zoom_in ( zoom\_factor: float = 1.25 ) \rightarrow None
```

Zoom into the current image.

**Parameters zoom\_factor** (*float*) – The new magnification factor for the image.

```
zoom_out ( zoom\_factor: float = 0.8 ) \rightarrow None
```

Zoom out of the current image.

**Parameters zoom\_factor** (*float*) – The new magnification factor for the image.

## $zoom_out_factor = 0.8$

The zoom-out factor used when the user zoom's out of the image via the zoom-out button.

class colourpaletteextractor.view.tabview.NewTab ( image\_id: Optional[str] = None,
image\_data: Optional[colourpaletteextractor.model.imagedata.ImageData] = None, parent=None )

Bases: PySide2.QtWidgets.QScrollArea

Modified QScrollArea to display and manipulate an image (via the ImageDisplay class).

**Parameters** •  $image_id(str)$  – The ID ('Tab\_xx') associated with a tab and image.

- image\_data (imagedata.ImageData) The ImageData object that hold the information associated with an image.
- parent Parent object of the NewTab Defaults to None.

#### image\_display

ImageDisplay used to show the QPixmap representation of the current image.

Type ImageDisplay

## ${\tt change\_toggle\_recoloured\_image\_pressed} \; (\;\;) \to None$

Toggle the \_toggle\_recoloured\_image\_pressed attribute between true and false (its opposite).

## property generate\_palette\_available: bool

The ability to generate the colour palette for the current NewTab object.

**Returns** (bool) – Returns true if the colour palette can be generated. Otherwise false.

## property generate\_report\_available: bool

The ability to generate the colour palette report for the current NewTab object.

**Returns (bool)** – Returns true if the colour palette report can be generated. Otherwise false.

## get\_slider\_positions() → PySide2.QtCore.QPointF

Get the grip positions of the horizontal and vertical scrollbars.

**Returns (QPointF)** – The position of the grip for the horizontal and veritcal scrollbars.

#### property image\_id: str

The image ID of the images and its data that is linked to the current NewTab object

**Returns (str)** – The ID ('Tab\_xx') associated with a tab and image.

## property progress\_bar\_value: float

The current level of progress shown by the status bar for the associated NewTab object.

**Returns (float)** – The current level of progress shown by thr status bar.

## **set\_slider\_positions** ( $x_position: float, y_position: float) <math>\rightarrow$ None

Set the position of the horizontal and vertical scrollbar's grip.

#### **Parameters**

- **x\_position** (*float*) Position of the grip for the horizontal scrollbar.
- **y\_position** (*float*) Position of the grip for the vertical scrollbar.

## staticMetaObject = <PySide2.QtCore.QMetaObject object>

## property status\_bar\_state: int

The current status bar state, represented by an integer.

See the otherviews.StatusBar.set\_status\_bar() method for more information.

**Returns (int)** – The current status bar state.

#### property toggle\_recoloured\_image\_available: bool

Stores the availability of the recoloured image (if it available to be displayed or not).

**Returns (bool)** – True if the recoloured image is available. Otherwise false.

## property toggle\_recoloured\_image\_pressed: bool

The status of the toggle button used to switch between the original image and the recoloured image.

**Returns (bool)** – True if the recoloured image is displayed by the GUI. Otherwise false.

wheelEvent ( event: PySide2.QtGui.QWheelEvent ) → None

Intercepts the super class' wheelEvent to allow zooming into and out of an image using the mousewheel.

Also calls the super class' wheelEvent handler at the end.

Parameters event (QWheelEvent) - Mousewheel event

## property zoom\_level: float

The degree of magnification for the currently displayed image.

**Returns** (float) – The degree of magnification for the current image.

Module contents

## 1.1.2 Module contents

- genindex
- modindex
- search

#### С

```
colourpaletteextractor, 22
   colourpaletteextractor.controller,
   colourpaletteextractor.controller.controller,
   colourpaletteextractor.controller.worker,
          1
   colourpaletteextractor.examples,
   colourpaletteextractor.examples.generatecolourpaletteexample,
   colourpaletteextractor.model, 12
   colourpaletteextractor.model.algorithms,
   colourpaletteextractor.model.algorithms.cielabcube,
   colourpaletteextractor.model.algorithms.nieves2020,
   colourpaletteextractor.model.algorithms.palettealgorithm,
          6
   colourpaletteextractor.model.generatereport,
   colourpaletteextractor.model.imagedata,
   colourpaletteextractor.model.model,
   colourpaletteextractor.tests, 13
   colourpaletteextractor.tests.helpers,
   colourpaletteextractor.tests.helpers.helperfunctions,
          12
   colourpaletteextractor.tests.nieves2020_test,
   colourpaletteextractor.view, 22
   colourpaletteextractor.view.mainview,
          13
   colourpaletteextractor.view.otherviews,
   colourpaletteextractor.view.tabview,
          19
```

#### el.imagedata.ImageData property), 9 \_close\_request\_action (colourpaletteextracapp\_icon (colourpaletteextractor.view.maintor.view.mainview.MainView view.MainView attribute), 15 attribute), 14 append\_title() (colourpaletteextractor.view.oth-\_max\_progress (colourpaletteextracerviews.ErrorBox method), 17 tor.view.otherviews.StatusBar attribute), 18 \_min\_progress (colourpaletteextractor.view.otherviews.StatusBar BatchGenerationProgressWidget (class in attribute), 18 colourpaletteextractor.view.otherviews), \_progress\_bar (colourpaletteextractor.view.otherviews.StatusBar attribute), 18 browse\_button (colourpaletteextrac-\_status\_label (colourpaletteextractor.view.othtor. view. otherviews. Preferences Widgeterviews. Status Bar attribute), 18 attribute), 17 Α C A4\_HEIGHT (colourpaletteextractor.model.-C\_STAR\_PERCENTILE (colourpaletteextracgeneratereport.ColourPaletteReport tor.model.algorithms.nieves2020.Nieves2020 attribute), 7 attribute), 4 A4\_WIDTH (colourpaletteextractor.model.genc\_stars (colourpaletteextractor.model.algoeratereport.ColourPaletteReport rithms.cielabcube.CielabCube properattribute), 7 about\_menu\_action (colourpaletteextraccalculate\_mean\_colour() (colourpaletteextractor.view.mainview.MainView tor.model.algorithms.cielabcube.CielabCube attribute), 14 method), 3 AboutBox (class in colourpaletteextraccancel\_batch\_button (colourpaletteextractor.view.otherviews), 16 tor. view. otherviews. Batch Generation Progress Widgetactive\_thread\_counter (colourpaletteextractor.attribute), 16 model. Model. Colour Palette Extractor Modelchange\_output\_directory() (colourpaletteextractor.model.model.ColourPaletteExtractorModel property), 10 add\_colour\_palette() (colourpaletteextracmethod), 10 tor.view.tabview.ColourPaletteDock change\_toggle\_recoloured\_image\_pressed() method), 19 (colourpaletteextractor.view.tabview.NewTab add\_image() (colourpaletteextractor.model.method), 21 model.ColourPaletteExtractorModel CielabCube (class in colourpaletteextractor.method), 10 model.algorithms.cielabcube), 3 add\_pixel\_to\_cube() (colourpaletteextractor.close\_current\_tab() (colourpaletteextrac-

model. algorithms. cielab Cube. Cielab Cube

erviews.PreferencesWidget attribute),

algorithm\_tab (colourpaletteextractor.view.oth-

18

algorithm\_used (colourpaletteextractor.mod-

tor.view.mainview.MainView

close\_temporary\_directory() (colourpaletteex-

tractor.model.model.ColourPaletteExtractorModel

method), 15

**Symbols** 

method), 10	module, 19
closeEvent() (colourpaletteextractor.view.main-	ColourPaletteExtractorController (class in
view.MainView method), 15	colourpaletteextractor.controller.controller),
colour_palette (colourpaletteextractor.mod-	
el.imagedata.ImageData property), 9	ColourPaletteExtractorModel (class in colour-
colour_palette_dock (colourpaletteextrac-	paletteextractor.model.model), 10
tor.view.mainview.MainView	ColourPaletteReport (class in colourpaletteex-
attribute), 14	tractor.model.generatereport), 7
colour_palette_relative_frequency (colourpalet-	continue_thread (colourpaletteextractor.mod-
teextractor.model.imagedata.ImageData	el.algorithms.palettealgorithm.PaletteAlgorithm
property), 9	property), 6
ColourBox (class in colourpaletteextrac-	continue_thread (colourpaletteextractor.mod-
tor.view.tabview), 19	el.imagedata.ImageData property), 9
ColourPaletteDock (class in colourpaletteex-	convert_lab_2_rgb() (in module colourpalette-
tractor.view.tabview), 19	extractor.model.algorithms.nieves2020),
colourpaletteextractor	5
module, 22	convert_rgb_2_lab() (in module colourpalette-
colourpaletteextractor.controller	extractor.model.algorithms.nieves2020),
module, 2	5
colourpaletteextractor.controller.controller	coordinates (colourpaletteextractor.model.algo-
module, 1	rithms.cielabcube.CielabCube proper-
colourpaletteextractor.controller.worker	ty), 3
module, 1	create_new_tab() (colourpaletteextractor.view
colourpaletteextractor.examples	mainview.MainView method), 15
module, 2	create_report() (colourpaletteextractor.model
colourpaletteextractor.examples.generatecolourpalett	
module, 2	method), 8
colourpaletteextractor.model	CUBE_SIZE (colourpaletteextractor.model.al-
module, 12	gorithms.nieves2020.Nieves2020
colourpaletteextractor.model.algorithms	attribute), 4
module, 7	current_tab_changed() (colourpaletteextractor
colourpaletteextractor.model.algorithms.cielabcube	controller.controller.ColourPaletteExtractorController
module, 3	method), 1
colourpaletteextractor.model.algorithms.nieves2020	
module, 4	D
colourpaletteextractor.model.algorithms.palettealgor	ithmFAULT ALGORITHM (colournaletteextrac-
module, 6	tor.model.model.ColourPaletteExtractorModel
colourpaletteextractor.model.generatereport	attribute), 10
module, 7	
colourpaletteextractor.model.imagedata	DEFAULT_HEIGHT (colourpaletteextractor
module, 8	model.model.ColourPaletteExtractorModel
colourpaletteextractor.model.model	attribute), 10
module, 10	default_new_tab_image (colourpaletteextrac-
colourpaletteextractor.tests	tor.view.mainview.MainView
	attribute), 15
module, 13	default_path_button (colourpaletteextrac-
colourpaletteextractor.tests.helpers	tor.view.otherviews.PreferencesWidget
module, 12	attribute), 17
colourpaletteextractor.tests.helpers.helperfunctions	DEFAULT_USE_USER_DIRECTORY (colour-
module, 12	paletteextractor.model.model.ColourPaletteExtractor
colourpaletteextractor.tests.nieves2020_test	attribute), 10
module, 13	DEFAULT_USER_DIRECTORY (colourpalette-
colourpaletteextractor.view	extractor.model.model.ColourPaletteExtractorModel
module, 22	
colourpaletteextractor.view.mainview	attribute), 10
module, 13	DEFAULT_WIDTH (colourpaletteextractor
colourpaletteextractor.view.otherviews	model.model.ColourPaletteExtractorModel
module, 16	attribute), 10
colourpaletteextractor.view.tabview	
corour pareticextractor, view, tabview	

_	tor.view.mainview.MainView
E	attribute), 14
elided_text() (colourpaletteextractor.view.oth-	generate_report_available (colourpaletteextrac-
erviews.ElidedLabel method), 17	tor.view.tabview.NewTab property), 21
ElidedLabel (class in colourpaletteextrac-	get_algorithms_and_buttons() (colourpaletteex-
tor.view.otherviews), 16	tractor.view.otherviews.PreferencesWidget method), 18
enterEvent() (colourpaletteextractor.view.tab-	get_c_star_percentile_value() (colourpaletteex-
view.ColourBox method), 19	tractor.model.algorithms.cielabcube.CielabCube
error (colourpaletteextractor.controller.worker	method), 3
WorkerSignals attribute), 2 ErrorBox (class in colourpaletteextrac-	get_c_stars() (in module colourpaletteextractor
tor.view.otherviews), 17	model.algorithms.nieves2020), 6
event() (colourpaletteextractor.view.tabview.Im-	get_image() (in module colourpaletteextrac-
ageDisplay method), 20	tor.tests.helpers.helperfunctions), 12
extension (colourpaletteextractor.model.image-	get_image_as_q_image() (colourpaletteextrac-
data.ImageData property), 9	tor.model.imagedata.ImageData static
	method), 9
F	get_image_data() (colourpaletteextractor.mod-
file_name_and_path (colourpaletteextractor	el.model.ColourPaletteExtractorModel
model.imagedata.ImageData proper-	method), 11 get_implemented_algorithms() (in module
ty), 9	colourpaletteextractor.model.algorithms.palettealgorith
finished (colourpaletteextractor.controller.work-	7
er.WorkerSignals attribute), 2	get_l_star_percentile_value() (colourpaletteex-
footer() (colourpaletteextractor.model.gener-	tractor.model.algorithms.cielabcube.CielabCube
atereport.ColourPaletteReport	method), 3
method), 7	get_relative_frequencies() (in module colour-
G	paletteextractor.model.algorithms.cielabcube),
	4
generate_all_palette_action (colourpaletteex-	get_settings() (in module colourpaletteextrac-
tractor.view.mainview.MainView attribute), 14	tor.model.model), 12 get_slider_positions() (colourpaletteextrac-
generate_all_report_action (colourpaletteex-	tor.view.tabview.NewTab method), 21
tractor.view.mainview.MainView	
attribute), 14	Н
generate_colour_palette() (colourpaletteextrac-	header (colourpaletteextractor.view.oth-
tor.model.algorithms.nieves2020.Nieves2020	erviews.ErrorBox attribute), 17
method), 5	header() (colourpaletteextractor.model.gener-
generate_colour_palette() (colourpaletteextrac-	atarapart Calaur Palatta Papart
tor.model.algorithms.palettealgorithm.Palett	teAlgorithmethod), 7
method), 6	
generate_colour_palette_from_image() (in	
module colourpaletteextractor.model model), 12	image (colourpaletteextractor.model.imageda-
generate_palette() (colourpaletteextractor.mod-	ta.ImageData property), 9
el.model.ColourPaletteExtractorModel	image_data_id_dictionary (colourpaletteextrac-
method), 11	tor. model. Model. Colour Palette Extractor Model
generate_palette_action (colourpaletteextrac-	property), 11
tor.view.mainview.MainView	image_display (colourpaletteextractor.view.tab-
attribute), 14	view.NewTab attribute), 21
generate_palette_available (colourpaletteextrac-	image_id (colourpaletteextractor.view.tab-
tor.view.tabview.NewTab property), 21	view.NewTab property), 21 IMAGE_START_POSITION (colourpaletteex-
generate_report() (colourpaletteextractor.mod-	tractor.model.generatereport.ColourPaletteReport
el.model.ColourPaletteExtractorModel	attribute), 7
method), 11	IMAGE_WIDTH (colourpaletteextractor.mod-
generate_report() (in module colourpaletteex- tractor.model.generatereport), 8	el.generatereport.ColourPaletteReport
generate_report_action (colourpaletteextrac-	attribute), 7
beriefate_report_action (colourpalettecking	

image_zoom() (colourpaletteextractor.view.tab-	colourpaletteextractor.tests.helpers.helperfunctions,
view.ImageDisplay method), 20 ImageData (class in colourpaletteextractor	colourpaletteextractor.tests.nieves2020_test,
model.imagedata), 8	13
ImageDisplay (class in colourpaletteextrac-	colourpaletteextractor.view, 22
tor.view.tabview), 20	colourpaletteextractor.view.mainview, 13
increment_pixel_count_after_reassignment()	colourpaletteextractor.view.otherviews, 16
(colour paletteextractor.model.algorithms.cie method), $4$	•
1	N
L	NAME (colourpaletteextractor.model.algo-
l_stars (colourpaletteextractor.model.algorithm-s.cielabcube.CielabCube property), 4	rithms.nieves2020.Nieves2020CentredCubes attribute), 5
label (colourpaletteextractor.view.otherviews	NAME (colourpaletteextractor.model.algo-
BatchGenerationProgressWidget	rithms.nieves2020.Nieves2020OffsetCubes
attribute), 16	attribute), 5
leaveEvent() (colourpaletteextractor.view.tab- view.ColourBox method), 19	name (colourpaletteextractor.model.algorithm- s.palettealgorithm.PaletteAlgorithm
	property), 6
M	name (colourpaletteextractor.model.imageda-
MainView (class in colourpaletteextrac-	ta.ImageData property), 9
tor.view.mainview), 13	NewTab (class in colourpaletteextrac-
MARGIN (colourpaletteextractor.model.gener-	tor.view.tabview), 20
atereport.ColourPaletteReport	Nieves2020 (class in colourpaletteextractor model.algorithms.nieves2020), 4
attribute), 7	Nieves2020CentredCubes (class in colourpalet-
MAX_IMAGE_HEIGHT (colourpaletteextrac-	to out the star me adal algorithms an invactor (2020)
tor.model.generatereport.ColourPaletteRepo	rt 5
attribute), 7 mean_colour (colourpaletteextractor.model.al-	Nieves2020OffsetCubes (class in colourpalette-
gorithms.cielabcube.CielabCube prop-	extractor.model.algorithms.nieves2020),
erty), 4	5
MIN_L_STAR (colourpaletteextractor.model.al-	
gorithms.nieves2020.Nieves2020	0
attribute), 4	open_action (colourpaletteextractor.view.main-
module	view.MainView attribute), 14
colourpaletteextractor, 22	output_tab (colourpaletteextractor.view.oth-
colourpaletteextractor.controller, 2	erviews.PreferencesWidget attribute),
colourpaletteextractor.controller.controller,	18
1	_
colourpaletteextractor.controller.worker, 1	P
colourpaletteextractor.examples, 2	paintEvent() (colourpaletteextractor.view.oth-
colourpaletteextractor.examples.generatecolourp	aletteexample erViews.ElidedLabel method), 17 PaletteAlgorithm (class in colourpaletteextrac-
colourpaletteextractor.model, 12	tor.model.algorithms.palettealgorithm),
colourpaletteextractor.model.algorithms, 7	
colourpaletteextractor.model.algorithms.cielabcu	prxel_count_after_reassignment (colourpalette-
	extractor.model.algorithms.cielabcube.CielabCube
colourpaletteextractor.model.algorithms.nieves20	DIODCILVI, I
4 colourpaletteextractor.model.algorithms.palettea	pixels (colourpaletteextractor.model.algorithm-
6	s.cielabcube.CielabCube property), 4
colourpaletteextractor.model.generatereport,	preferences_menu_action (colourpaletteextrac-
7	tor.view.mainview.MainView
colourpaletteextractor.model.imagedata, 8	attribute), 14 Professor assWidget (class in colournal attracy
colourpaletteextractor.model.model, 10	PreferencesWidget (class in colourpaletteex-
colourpaletteextractor.tests, 13	tractor.view.otherviews), 17 progress (colourpaletteextractor.controller
colourpaletteextractor.tests.helpers, 12	worker.WorkerSignals attribute), 2

progress_bar_value (colourpaletteextrac-	show_output_directory_dialog_box() (colour-
tor.view.tabview.NewTab property), 21	paletteextractor.view.otherviews.PreferencesWidget
	method), 18
R	show_palette_dock_action (colourpaletteextrac-
read_view_settings() (colourpaletteextractor	tor.view.mainview.MainView
model.model.ColourPaletteExtractorModel	attribute), 15
static method), 11	show_preferences() (colourpaletteextrac-
recoloured_image (colourpaletteextractor.mod-	tor.view.otherviews.PreferencesWidget
el.imagedata.ImageData property), 9	method), 18
relevant (colourpaletteextractor.model.algo-	show_toolbar_action (colourpaletteextrac-
rithms.cielabcube.CielabCube proper-	tor.view.mainview.MainView
ty), 4	attribute), 15
remove_colour_palette() (colourpaletteextrac-	show_widget() (colourpaletteextrac-
tor.view.tabview.ColourPaletteDock	tor. view. other views. Batch Generation Progress Widget
method), 20	method), 16
remove_image_data() (colourpaletteextractor	sort_colour_palette() (colourpaletteextractor
model.model.ColourPaletteExtractorModel	model.imagedata.ImageData method),
method), 11	9
ReportGenerator (class in colourpaletteextrac-	staticMetaObject (colourpaletteextractor.con-
tor.model.generatereport), 8	troller.worker.WorkerSignals
RESOURCES_DIR (colourpaletteextrac-	attribute), 2
tor.view.mainview.MainView	staticMetaObject (colourpaletteextractor.view
attribute), 15	mainview.MainView attribute), 16
resources_path (colourpaletteextractor.view	staticMetaObject (colourpaletteextrac-
mainview.MainView attribute), 15	tor.view.otherviews.AboutBox
result (colourpaletteextractor.controller.work-	attribute), 16
er.WorkerSignals attribute), 2	staticMetaObject (colourpaletteextrac-
run() (colourpaletteextractor.controller.worker	tor.view.otherviews.BatchGenerationProgressWidget
Worker method), 2	attribute), 16
Worker method), 2	staticMetaObject (colourpaletteextrac-
S	tor.view.otherviews.ElidedLabel
	attribute), 17
save_report() (colourpaletteextractor.model	staticMetaObject (colourpaletteextrac-
generatereport.ReportGenerator	tor.view.otherviews.ErrorBox
method), 8	attribute), 17
SECONDARY_THRESHOLD (colourpaletteex-	staticMetaObject (colourpaletteextrac-
tractor.model.algorithms.nieves2020.Nieves2	tor.view.otherviews.PreferencesWidget
attribute), 4	attribute), 18
set_algorithm() (colourpaletteextractor.model	staticMetaObject (colourpaletteextrac-
model.ColourPaletteExtractorModel	tor.view.otherviews.StatusBar
method), 12	attribute), 19
set_cancel_text() (colourpaletteextrac-	staticMetaObject (colourpaletteextrac-
tor.view.otherviews.BatchGenerationProgres	ssWidget tor.view.tabview.ColourBox attribute),
method), 16	19
set_progress_callback() (colourpaletteextrac-	staticMetaObject (colourpaletteextrac-
tor.model.algorithms.palettealgorithm.Palet	teAlgorithmr.view.tabview.ColourPaletteDock
method), 6	attribute), 20
set_slider_positions() (colourpaletteextrac-	
tor.view.tabview.NewTab method), 21	staticMetaObject (colourpaletteextrac-
	tor.view.tabview.ImageDisplay
set_status_bar() (colourpaletteextrac-	tor.view.tabview.ImageDisplay attribute), 20
tor.view.otherviews.StatusBar	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac-
tor.view.otherviews.StatusBar method), 19	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21
tor.view.otherviews.StatusBar method), 19 show_file_dialog_box() (colourpaletteextrac-	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21 status (colourpaletteextractor.view.mainview
tor.view.otherviews.StatusBar method), 19 show_file_dialog_box() (colourpaletteextrac- tor.view.mainview.MainView	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21 status (colourpaletteextractor.view.mainview MainView attribute), 15
tor.view.otherviews.StatusBar method), 19 show_file_dialog_box() (colourpaletteextrac- tor.view.mainview.MainView method), 16	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21 status (colourpaletteextractor.view.mainview MainView attribute), 15 status_bar_state (colourpaletteextrac-
tor.view.otherviews.StatusBar method), 19 show_file_dialog_box() (colourpaletteextrac- tor.view.mainview.MainView method), 16 show_help_action (colourpaletteextrac-	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21 status (colourpaletteextractor.view.mainview MainView attribute), 15 status_bar_state (colourpaletteextrac- tor.view.tabview.NewTab property), 21
tor.view.otherviews.StatusBar method), 19 show_file_dialog_box() (colourpaletteextrac- tor.view.mainview.MainView method), 16	tor.view.tabview.ImageDisplay attribute), 20 staticMetaObject (colourpaletteextrac- tor.view.tabview.NewTab attribute), 21 status (colourpaletteextractor.view.mainview MainView attribute), 15 status_bar_state (colourpaletteextrac-

stop_action (colourpaletteextractor.view.main- view.MainView attribute), 14	toggle_recoloured_image_available (colour-paletteextractor.view.tabview.NewTab
SUPPORTED_IMAGE_TYPES (colourpaletteex- tractor.model.model.ColourPaletteExtractorNattribute), 10	property), 21  Modgle_recoloured_image_pressed (colourpalet- teextractor.view.tabview.NewTab
Т	property), 21 tools (colourpaletteextractor.view.mainview
tabs (colourpaletteextractor.view.mainview MainView attribute), 13	MainView attribute), 15
test_closest_relevant_colour_used_to_recolour_pixel()	JU
(in module colourpaletteextractor.test-	update_image() (colourpaletteextrac-
s.nieves2020_test), 13	tor.view.tabview.ImageDisplay
test_cube_colour_must_occur_more_than_three_perc	
(in module colourpaletteextractor.test-s.nieves2020_test), 13	update_preferences() (colourpaletteextrac- tor.view.otherviews.PreferencesWidget
test_cube_colour_must_occur_more_than_three_perc	
(in module colourpaletteextractor.test-	update_progress() (colourpaletteextrac-
s.nieves2020_test), 13 test_low_a_b_colour_does_not_meet_secondary_requ	tor.view.otherviews.BatchGenerationProgressWidget
(in module colourpaletteextractor.test-	update_progress_bar() (colourpaletteextrac-
s.nieves2020_test), 13	tor.view.otherviews.StatusBar
test_low_a_b_colour_does_not_meet_secondary_requ	
(in module colourpaletteextractor.test-	URL (colourpaletteextractor.model.algorithm-
s.nieves2020_test), 13	s.nieves2020.Nieves2020CentredCubes
test_nieves2020_centred_cubes_constructor()	attribute), 5
(in module colourpaletteextractor.test-	URL (colourpaletteextractor.model.algorithm-
s.nieves2020_test), 13	s.nieves2020.Nieves2020OffsetCubes
test_nieves2020_offset_cubes_constructor() (in	attribute), 5
module colourpaletteextractor.test-	url (colourpaletteextractor.model.algorithms
s.nieves2020_test), 13	palettealgorithm.PaletteAlgorithm
test_primary_requirements_1() (in module	property), 7
colourpaletteextractor.tests.nieves2020_test), 13	tor.view.otherviews.PreferencesWidget
test_primary_requirements_2() (in module	attribute), 18
colourpaletteextractor.tests.nieves2020_test),	
13	tor.view.otherviews.PreferencesWidget
test_primary_requirements_3() (in module	attribute), 17
colourpaletteextractor.tests.nieves2020_test),	
13	W
test_recoloured_image_of_same_size_1() (in	wheelEvent() (colourpaletteextractor.view.tab-
module colourpaletteextractor.test-	view.NewTab method), 22
s.nieves2020_test), 13	Worker (class in colourpaletteextractor.con-
test_recoloured_image_two_colours_1() (in module colourpaletteextractor.test-	troller.worker), 1
s.nieves2020_test), 13	WorkerSignals (class in colourpaletteextractor
test_recoloured_image_two_colours_2() (in	controller.worker), 2
module colourpaletteextractor.test-	write_default_settings() (colourpaletteextrac-
	tor.model.model.ColourPaletteExtractorModel
s.nieves2020_test), 13 test_two_colours_in_same_cube_can_meet_secondary (in_module_colournaletteextractor_test-	threshold 1() (colournaletteeytractor -
(in module colourpaletteextractor.test-	model.model.ColourPaletteExtractorModel
s.nieves2020_test), 13	static method), 12
THRESHOLD (colourpaletteextractor.model.al-	
gorithms.nieves2020.Nieves2020	Z
attribute), 4	zoom_factor (colourpaletteextractor.view.tab-
toggle_recoloured_image_action (colourpalet-	view.ImageDisplay attribute), 20
teextractor.view.mainview.MainView attribute), 14	zoom_in() (colourpaletteextractor.view.tab-
amputej, 17	view.ImageDisplay method), 20

zoom\_in\_action (colourpaletteextractor.view.mainview.MainView attribute), 14
zoom\_level (colourpaletteextractor.view.tabview.NewTab property), 22
zoom\_out() (colourpaletteextractor.view.tabview.ImageDisplay method), 20
zoom\_out\_action (colourpaletteextractor.view.mainview.MainView attribute), 14
zoom\_out\_factor (colourpaletteextractor.view.tabview.ImageDisplay
attribute), 20