

Une liseuse pour les passionnés de BDs

Comic Book Reader/Writer

Yufei HU
7 mars 2021

CONTENU

sommaire



01

Introduction

02

Structure

03

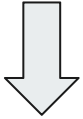
Fonctionnement

04

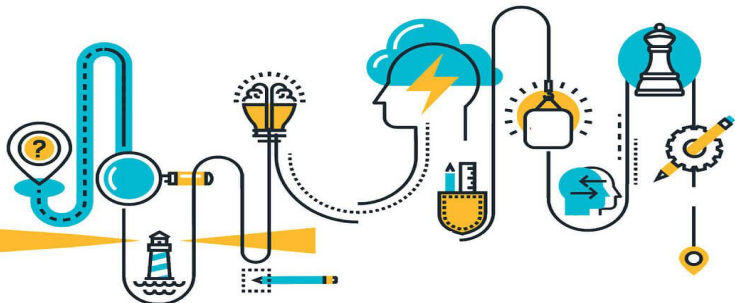
Conclusion

Introduction

Requirements

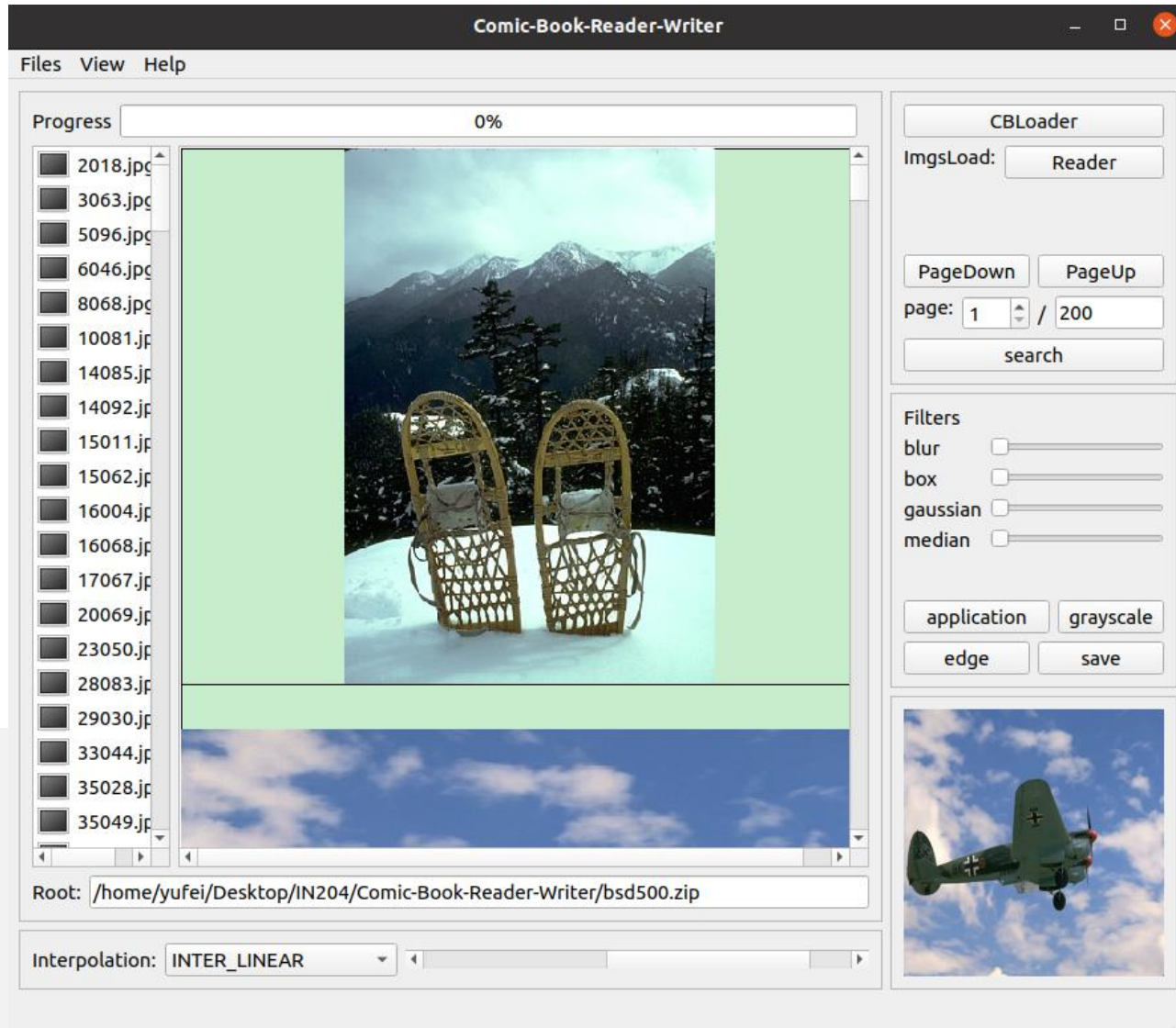


- quazip-1.1
 - OpenCV-4.5.1
 - Magick
 - QT Creator
-



01 Introduction

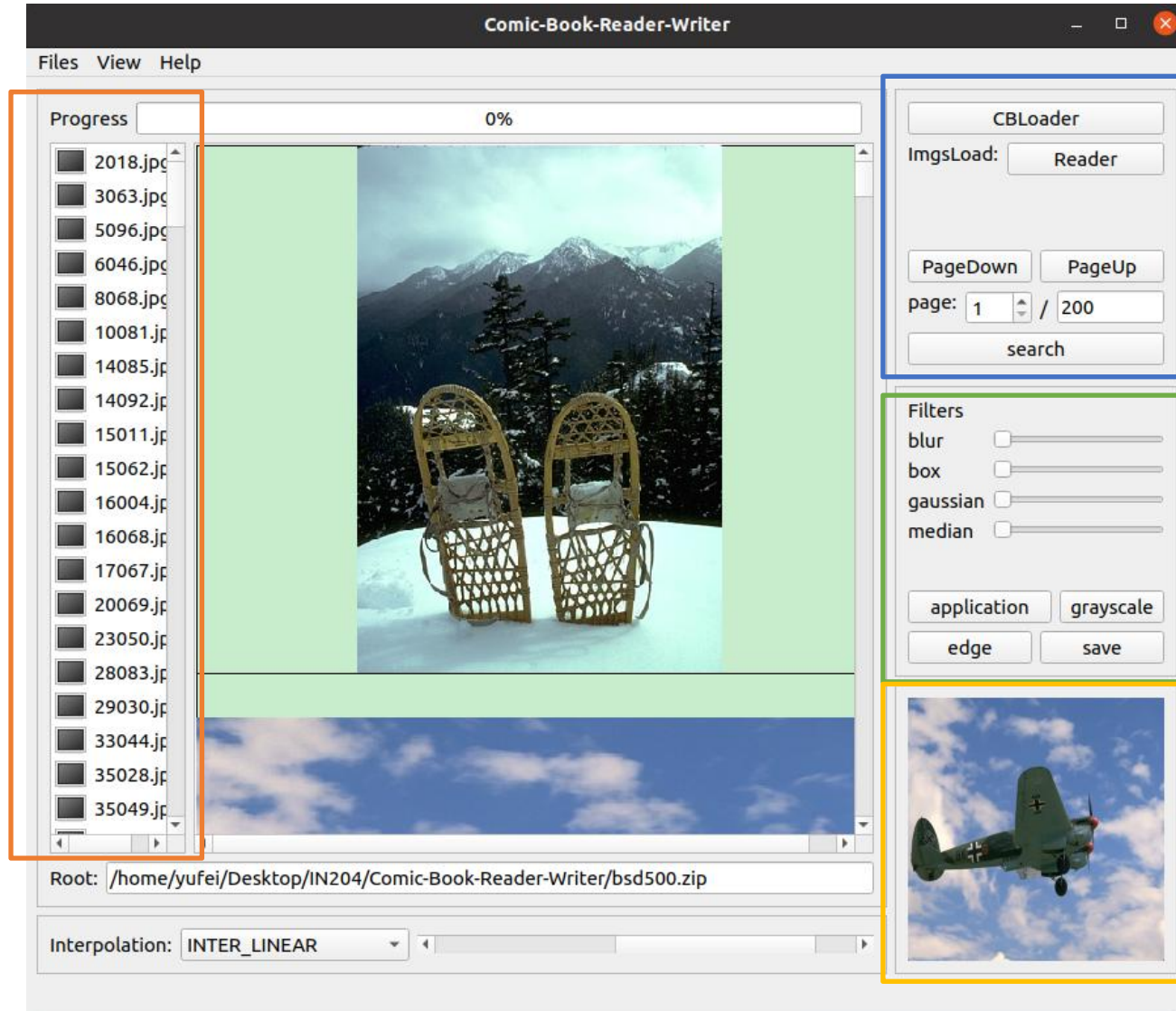
interface



01 Introduction

interface

table des matières



Contrôle de charge

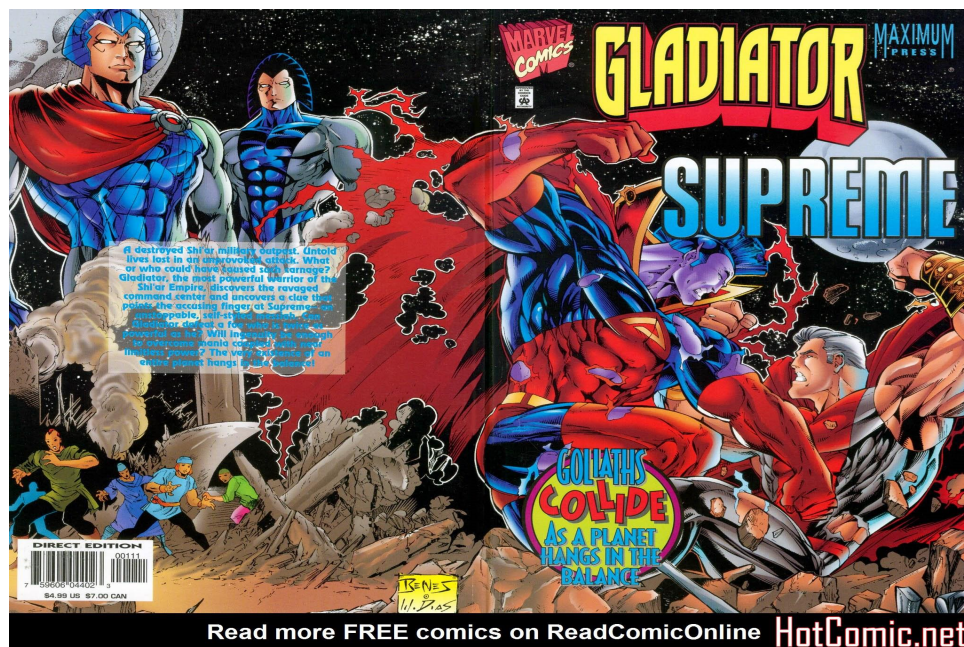
Filtre de traitement
d'image

Zone d'affichage
auxiliaire

01 Introduction

data

L'un des ensembles de données de cet article provient de l'open source en ligne et l'autre est un ensemble de test de ¹BSD500.



02 Structure de classe

archivemanager

ArchiveManager

```
QString getSrcPath();  
QString getDstPath();  
void setSrcPath(QString path);  
void setDstPath(QString path);  
  
virtual void archiveStartExtract(const QString&)=0;  
//list the contents of a zip file  
virtual void ListContents(const QString&)=0;
```

dérivation

hériter

ZipManager

```
ZipManager(){  
virtual~ZipManager();  
  
void ZipCompressDir(QString ZipFile,QString DirPath);  
void ZipCompressFiles(QString ZipFile,QStringList Files);  
QStringList ZipDecompressDir(QString ZipFile,QString DirPath);  
QStringList ZipDecompressFiles(QString ZipFile,QString DirPath,QStringList Files);  
void ListContents(const QString &);  
void archiveStartExtract(const QString &Path);
```

RarManager

02 Structure de classe

browsermanager

browserManager

```
browserManager();  
~browserManager();  
int getComicPage();  
void setComicPage(const int p);  
void setPageMaximum(const int p);  
int getPageMaximum();  
void setPageMinimum(const int p);  
int getPageMinimum();  
  
private:  
    int page;  
    int pageMaximum;  
    int pageMinimum;
```


02 Structure de classe

imgmanager

ImgManager

```
explicit ImgManager( QLabel *qlabel, QWidget *parent = 0);
explicit ImgManager(QWidget *parent = 0);
void img_display_on_label(const QImage img);

QImage getImg(const QString& srcPath);

QImage loadImg(const QString& srcPath);

QImage update(const QString &srcPath);

QImage keepOriginalImage(const QString &srcPath);

void on_boxFilter(const int kernel_size);

void on_blur(const int kernel_seze);

void on_gaussianBlur(const int kernel_size);

void on_medianBlur(const int kernel_size);

QImage on_grayscale(const QString &srcPath);

QImage on_egdeDetection(const QString &srcPath);

void on_changementSave(const QString &srcPath);

void on_imgsSaved2PDF();

QImage on_Interpolation_zoom(const QString &srcPath, const int factor, const int mode);

private:
    QLabel *img_display_region;
    cv::Mat srcImg;
    cv::Mat dstImg;
    QString imgType;
```

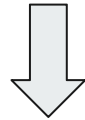
02 Structure de classe

mylistwidgetdelegate

¹myListWidgetDelegate

```
private:
    //define the way of paint
    void paint(QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index) const;

    //define the size of unit grill
    QSize sizeHint(const QStyleOptionViewItem &option, const QModelIndex &index) const;
};
```



Personnalisez QListWidget pour un meilleur affichage de QIcon (images).

02 Structure de classe

mainwindow

MainWindow

```
void on_actionOpenFiles_triggered();  
  
void on_loadComicBook_clicked();  
  
void on_pushButtonImgReader_clicked();  
  
void on_initial_listwidget_items();  
  
QImage onImgShow(const QString &img_path);  
  
void on_pushButton_page_next_clicked();
```

```
void on_pushButton_page_prev_clicked();  
  
void on_horizontalSlider_blur_valueChanged(int value);  
  
void on_horizontalSlider_box_valueChanged(int value);  
  
void on_horizontalSlider_gaussian_valueChanged(int value);  
  
void on_horizontalSlider_median_valueChanged(int value);
```

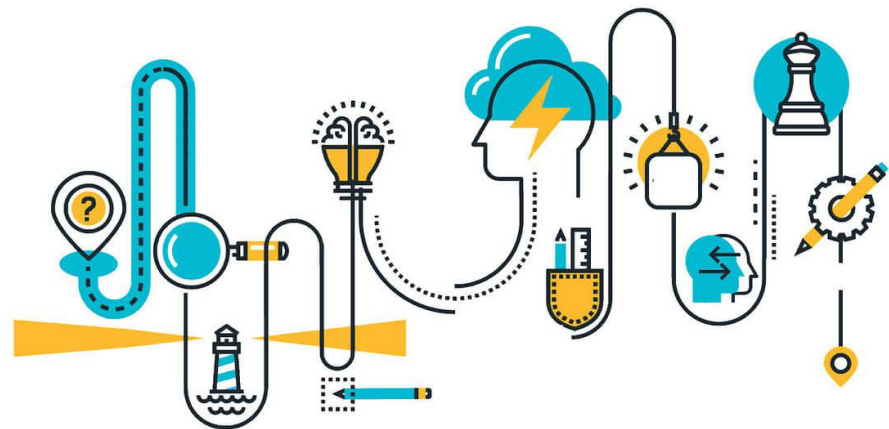
```
void on_bt_grayscale_clicked();  
  
void on_bt_edgeDetection_clicked();  
  
void on_bt_save_changement_clicked();  
  
void on_bt_page_search_clicked();
```

```
void on_horizontalScrollBar_for_zoom_sliderReleased();  
  
void on_actionsave_pdf_triggered();  
  
void on_listViewForContent_doubleClicked(const QModelIndex &index);
```

03 Fonctionnement

Liseuse de Bande-Dessinées[3]

- Formats fichiers de départ(zip)
- Formats fichiers images(jpg, png, bmp)
- Filtrages numérique(Filtre gaussien, filtre médian, filtre moyen, filtre boîte)
- Traitement d'image(niveaux de gris et détection de contour)
- Page suivante, précédente et recherche de page
- La barre de progression montre la progression de la lecture
- Enregistrez les images et enregistrez-les en version PDF
- L'écran secondaire affiche les informations sur la page suivante
- Le moniteur principal peut afficher deux images en même temps et dispose d'un mode de protection des yeux vert
- L'affichage principal peut être glissé pour tourner les pages
- Afficher le chemin du fichier
- Le répertoire de données actuel et vous pouvez double-cliquer pour tourner la page
- Peut continuer à se développer et a un certain degré de tolérance aux pannes



04 Conclusion

En conclusion:

- Ce projet peut essentiellement répondre aux exigences du projet, et en même temps, il peut être re-développé en héritant et en ajoutant des algorithmes de traitement d'image associés.
- Ce projet a été développé sous linux et n'a pas réussi à ajouter des tâches de décompression bit7z et rar. Ce projet ne prend en charge que les fichiers zip.
- [application.gif](#)

Merci pour votre attention.

Bibliothèques

- [1] [youtube]<https://www.youtube.com/watch?v=mxlcKmvMK9Q>
- [2] [github]<https://github.com/Longxr/ComicReader>.
- [3] [IN204]<https://perso.ensta-paris.fr/~bmonsuez/Cours/doku.php?id=in204:resources>.

