FACE REPLACE (GOOGLE API)

Documentazione Google Vision Api:
Link - https://cloud.google.com/vision/docs/detecting-faces

- Creiamo un nuovo *Job*:

```
php artisan make:job GoogleVisionRemoveFaces
```

- Nel *job* appena creato:

```
use Spatie\Image\Image;
use Spatie\Image\Manipulations;

class GoogleVisionRemoveFaces implements ShouldQueue{
      use Dispatchable ,InteractsWithQueue, Queueable, SerializeModels;

      private $announcement_image_id;

      public function __construct($announcement_image_id){
            $this->announcement_image_id = $announcement_image_id;
      }
      public function handle(){
            $i = AnnouncementImage::find($this->announcement_image_id);
            if (!$i){
                  return;
            }
            $srcPath = storage_path('/app/' . $i->file);
            $image = file_get_contents($srcPath);

            putenv('GOOGLE_APPLICATION_CREDENTIALS=' .
                                                base_path('google_credential.json'));
      $imageAnnotator = new ImageAnnotatorClient();
      $response = $imageAnnotator->faceDetection($image);
      $faces = $response->getFaceAnnotations();

      foreach($faces as $face){
            $vertices = $face->getBoundingPoly()->getVertices();

            $bounds = [];

            foreach($vertices as $vertex){
                  $bounds[] = [$vertex->getX(), $vertex->getY()];
            }

            $w = $bounds[2][0] - $bounds[0][0];
            $h = $bounds[2][1] - $bounds[0][1];

            $image = Image::load($srcPath);
            $image->watermark(base_path('resources/img/smile.png'))
                  ->watermarkPosition('top-left')
```

```
                   ->watermarkPadding($bounds[0][0], $bounds [0][1])
                   ->watermarkWidth($w, Manipulations::UNIT_PIXELS)
                   ->watermarkHeight($h, Manipulations::UNIT_PIXELS)
                   ->watermarkFit(Manipulations::FIT_STRETCH);
             $image->save($srcPath);
      }
      $imageAnnotator->close();
}
```

- Aggiorno il la *funzione* di creazione dell’annuncio in modo che i *dispatch* siano dati con l’ordine corretto. Le immagini devono avere la copertura del volto con l’immagine prestabilita nel *Job* prima dei *resize*.

```
foreach($images as $image){
      $i = new AnnouncementImage();

      $fileName = basename($image);
      $newFileName = "public/announcements/{$a->id}/{$fileName}";

      Storage::move($image, $newFileName);

      $i->file = $newFileName;
      $i->announcement_id = $a->id;

      $i->save();

      GoogleVisionSafeSearchImage::withChain([
            new GoogleVisionLabelImage($i->id),
            new GoogleVisionRemoveFaces($i->id),
            new ResizeImage($i->file, 300, 150),
            new ResizeImage($i->file, 400, 300)
      ])->dispatch($i->id);
}
```

NB. Ricordiamoci di chiudere e riaprire il *php artisan queue:work*

NB. Possiamo aprire piu’ *work* contemporaneamente per avere una processazione piu’ veloce dato che i *processi* verranno automaticamente suddivisi.

General error: 11 database disk image is malformed

Nella *bash*:

```
php artisan scout:flush "App\Models\Article"
php artisan scout:import "App\Models\Article"
```