

RAPPORT DE STAGE - TN09 - GI

Développement de l'interface homme-machine pour l'inspection automatisée

13 FÉVRIER 2023 - 28 JUILLET 2023

Auteur :
Zhentao XU

Encadrant UTC :
Véronique CHERFAOUI

Encadrant entreprise :
Julien MONVILLE

Entreprise :
Deltacad
795 rue de Longues Rayes
60610, Lacroix Saint-Ouen

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, je tiens à remercier vivement Vincent ARGENTON, mon maitre de stage, pour le temps qu'ils m'a toujours accordé sans hésitation, ainsi que pour les opportunités et responsabilités qu'il m'a offertes.

Ensuite, je voudrais remercier le manager de l'équipe, Stéphane GLOAGUEN, pour la reconnaissance, le temps et les conseils qu'il m'a accordé.

De plus, je souhaite aussi remercier Kevin HELOUART, qui, avec Vincent, a su m'intégrer à l'équipe très rapidement.

Enfin, je remercie Michelangelo NERI et Milan RADOVIC d'avoir accepté ma candidature.

Je souhaite aussi remercier tout mes collègues pour le temps passé à leurs côtés, ainsi que pour leurs précieux conseils, et plus spécialement Thibault, Bastien et Dominique.

Résumé technique

Mon stage se déroule à Deltacad, une société d'ingénierie spécialisée en informatique scientifique. En collaboration avec une entreprise AML System et deux instituts de recherche, les laboratoires Roberval et Heudiasyc de l'UTC, nous développons un logiciel pour des scénarios industriels, qui entraîne des modèles d'apprentissage automatique à partir d'images de pièces sur la chaîne de production et les applique pour déterminer la qualité des pièces et détecter les défauts.

Dans ce cadre, ma tâche consistait à concevoir et à développer l'interface homme-machine et à améliorer progressivement sa fonctionnalité en fonction des besoins. Durant mon stage, J'ai réalisé une interface du logiciel extensible en utilisant WPF et j'ai conçu l'architecture globale du logiciel, en intégrant des modules fonctionnels déjà conçus ainsi que certains de mes propres modules fonctionnels. Par ailleurs, j'ai également utilisé certaines extensions externes, telles que l'affichage des journaux d'opérations via le cadre POA (programmation orientée aspect), et l'enregistrement et l'importation des informations de configuration via la sérialisation xml.

Mots-clés : .Net WPF, MVVM(Model-View-ViewModel), C#, POA, python, C++

Introduction

Dans le cadre de ma formation d'ingénieur à l'Université de Technologie de Compiègne, le troisième semestre est consacré à un stage de 24 semaines d'assistant ingénieur. Cette expérience comme une longue période de travail professionnel, nous permet d'avoir un aperçu du fonctionnement de l'entreprise et du monde professionnel, d'appliquer les connaissances acquises lors de sa formation, d'exprimer notre créativité et nos idées jeunes et énergiques. Et dans ce rapport de stage, je résume le travail que j'ai effectué à Deltacad au cours des six mois allant de février 2023 à juillet 2023 et je le présente plus en détail en trois partis principaux.

Dans la première partie, je vous donnerai une vue d'ensemble de mon entreprise et mon point de vue sur mon équipe. Cela vous donnera un premier aperçu de l'environnement de mon stage.

Dans la deuxième partie, je vous présenterai quelques détails sur l'ensemble de mon stage, y compris le sujet du stage, l'organisation des tâches, ma contribution et les aspects techniques du projet. En résumé, cette section est plus orientée vers une vision globale et n'est pas divisée en modules individuels.

Dans la troisième section, je décomposerai mes réalisations au cours de mon stage en un certain nombre de points afin de les expliquer plus en détail. Je également expliquerai les raisons pour lesquelles j'ai choisi certaines options, ainsi que les difficultés et solutions techniques rencontrées à ce moment-là. Dans cette partie, je me concentrerai sur les aspects techniques et, en raison de la spécificité du projet, je parlerai des différents modules du projet pour vous donner une compréhension plus claire.

Enfin, je résumerai les améliorations que j'ai apportées pendant mon stage et je donnerai quelques indications sur le développement futur du projet.

1. Présentation de Deltacad

1.1 Le groupe Deltacad

1.1.1 Historique du groupe

2. Mission

2.1 Sujet de stage

Le sujet de mon stage était initialement Développement/Intégration de fonctions de Vision par ordinateur pour l'inspection automatisée. Durant toute la période de mon stage, mon travail s'est axé autour d'un logiciel appelé **PowerEye**. **PowerEye** est un logiciel développé pour des scénarios industriels qui utilise la vision par ordinateur pour détecter les défauts des pièces sur la chaîne de production. Les laboratoires de l'UTC faisant partie de ce projet, ils assurent le développement des machines apprenantes. Deltacad est responsable de la conception de l'interface de l'ensemble du projet et de l'intégration des autres fonctions.

Avant que je ne commence mon stage, **PowerEye** avait déjà été conçu avec certaines fonctionnalités ainsi qu'un prototype d'interface. Cependant, il était clair que le système n'était pas encore adapté à une application formelle au sein de l'entreprise. Par conséquent, la tâche principale de mon stage consistait à concevoir et améliorer une interface homme-machine plus pratique basée sur le prototype de l'interface et essayer de tester les pièces d'AML Systems¹ à l'aide de cette interface après intégration de la fonction.

En ce qui concerne les lignes directrices spécifiques pour la conception de l'interface, le chef de projet et le suiveur m'ont fourni un aperçu général des différents modules qu'elle contiendrait et des fonctionnalités qu'elle pourrait inclure. Après avoir confirmé le thème et le programme de travail, mon stage a officiellement commencé.

1. une société du segment éclairage du groupe Johnson Electric, conçoit, produit et commercialise, des solutions pour améliorer la visibilité, la sécurité, et le confort du conducteur

2.2 Planning de stage

Au début de mon stage, mes tâches étaient planifiées en trois phases :

- **1.** Développement de l'interface homme-machine de PowerEye
- **2.** Test sur les données de l'entreprise partenaire AML système
- **3.** Déploiement d'applications dans le système AML

Dans le cadre du travail réel, pour chaque phase de la tâche, je l'ai décomposée en différentes étapes. La première phase peut en fait être divisée en plusieurs parties : la conception de l'interface, la réalisation de l'apparence de l'interface, la réalisation de la structure des données de l'interface et la réalisation des fonctions de la couche d'application de l'interface. La deuxième phase comprend également l'intégration de la fonctionnalité conçue, le test des données et l'amélioration de la fonctionnalité. Il est très regrettable qu'en raison de contraintes de temps, mon stage se soit terminé à la deuxième phrase. Mais en fait, en plus des tâches incluses dans le plan, j'ai également complété un certain nombre d'autres fonctionnalités, y compris enregistrement des appels de fonction et exécution de programmes de script `c#` via des chaînes de caractères

À mon avis, la partie la plus importante du processus de stage est la structure globale du projet, car elle influe grandement sur le développement des idées et l'évolutivité du projet. À cette fin, j'ai amélioré l'architecture globale à plusieurs reprises au cours du processus de développement, en utilisant des `prism` au début de projet individuel, en divisant l'interface en couches d'affichage et d'application, et en développant avec le cadre `mvvm`.

2.3 Contributions

Comme mentionné dans la section précédente, lorsque j’ai commencé à travailler sur le projet, il existait déjà un prototype d’interface ainsi que certains modules fonctionnels.

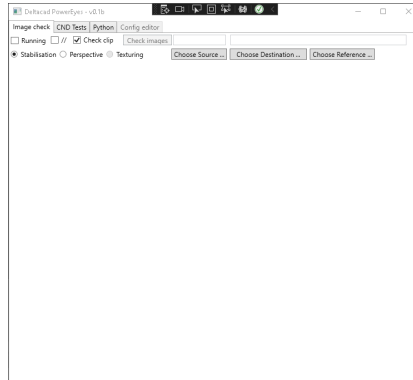


FIGURE 1 – Prototype d’interface PowerEye

L’interface se compose de quatre modules : Image check, CND Tests, python et config editor. Il contient des méthodes d’application de Template Matching pour la détection des pièces et des modèles d’entraînement pour les tests avec des ensembles de données. Mais le fait est que l’interface prototype n’est pas suffisante pour inclure toutes les fonctionnalités envisagées par PowerEye. Par conséquent, ma principale contribution tout au long du stage a été la conception et le développement d’une nouvelle version de l’interface PowerEye. À la fin du stage, la nouvelle version de l’interface de powereye ressemblait à ceci.

Il contient cinq modules :

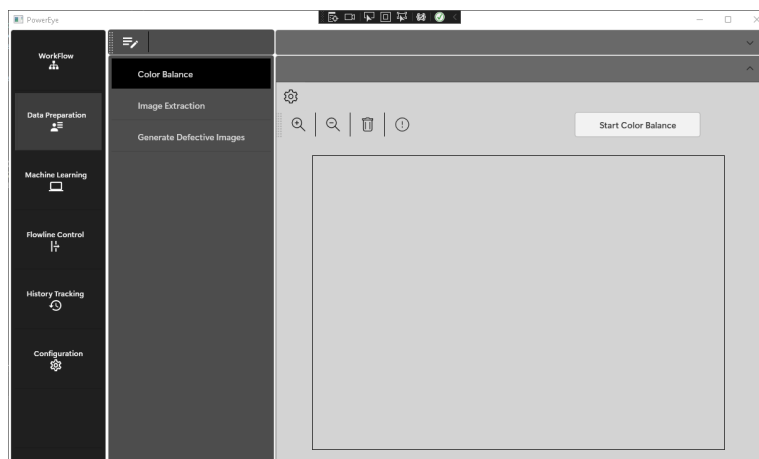


FIGURE 2 – Interface PowerEye redessinée

- **Préparation des données (Data preparation)** : Ce module permet d’extraire les contours des pièces de l’image et d’ajouter du bruit de perlin à l’image.

- **Apprentissage automatique (machine learning)** : Ce module est utilisé pour entraîner le modèle ckpt à partir de l'ensemble de données.
- **Contrôle de la chaîne d'assemblage (flowline control)** : Ce module détecte si une pièce est qualifiée par Template Matching, et enregistre les données du résultat du test.
- **Interroger les données historiques (History tracking)** : Ce module interroge les données du fichier Excel par mots-clés.
- **Configuration des données globales (configuration)** : Ce module affiche la configuration complète du logiciel en cours et l'importe par l'intermédiaire de fichiers de configuration.

En plus de la partie principale du logiciel, j'ai mis en place les fonctionnalités suivantes

- Utilisation du cadre AOP pour afficher les actions de la couche applicative.
- Implémentation d'un script c# qui s'exécute à partir d'une chaîne de caractères.
- Implémentation d'ajouts manuels de fonctionnalités existantes pour personnaliser le flux de travail.

Ce qui ci-dessus résume essentiellement toutes les contributions que j'ai apportées au cours de mon stage. Dans mon travail, j'ai conçu et développé le projet de manière indépendante en utilisant les exigences comme point de départ, et j'ai finalement bien accompli la tâche.

2.4 Outils et technologies

Tout au long de mon stage, j'ai travaillé avec les outils suivants :

— **Pour le développement :**

- *Comme langage de programmation* : C#, WPF.NET, C++, Python
- *Comme environnement de développement* : Microsoft Visual Studio, **IDE** complet pour les développeurs .NET et C++ sur Windows.

— **Pour la bureautique** : **LaTeX**, pour la rédaction de ce rapport. WebEx, pour les réunions de projet à distance

— **Pour la gestion des différentes versions du logiciel** : **SmartSVN**, un système de gestion des versions.

Pour la réalisation du projet, j'ai utilisé les extensions suivantes :

- **WPF-UI** : Extension pour modifier le style du contrôle
- **MahApps.Iconpacks** : Extension pour l'ajout d'icônes de boutons
- **Fody** : Extension pour la mise en œuvre du cadre aop
- **YamlDotNet** : Extension pour les modifications de l'édition yaml

2.5 Validation de travaux

La validation de mes travaux se déroulait lors de réunions avec l'équipe de projet. Comme le chef de projet ne travaille pas au même endroit que moi, nos réunions se passaient généralement à distance via le WebEx.

Au cours de la réunion, normalement, je commençais par présenter et démontrer l'état d'avancement de la tâche. Ensuite, le tuteur et le chef de projet discutaient avec moi des améliorations à apporter et planifient les tâches suivantes. le contenu de chaque réunion était enregistré par le chef de projet et m'était envoyé par e-mail après la réunion.

Cela me permet de partager mon point de vue sur le projet et me donne une idée claire de ce qui doit être fait a la phase actuel, ce qui améliore grandement mon efficacité et standardise mon travail.

2.6 Prise de recul

2.6.1 Intérêt de mon travail

L'intérêt de mon travail pour l'équipe était de concevoir et développer une IHM plus aboutie que la précédente, migrer et améliorer les fonctionnalités développées précédemment. En outre, le cadre aop et le modèle de l'observateur sont plus propices à la maintenance de l'interface, et il existe un bon cadre général pour le développement du suivi.

2.6.2 Améliorations à envisager

Mais tout comme le développement de nombreux logiciels connus nécessite beaucoup de polissage, le logiciel que j'ai développés au cours de mon stage de six mois n'étaient certainement pas parfait. Dans le développement futur du projet, les améliorations suivantes sont nécessaires.

- Développer la fonctionnalité de certains modules et les intégrer dans l'interface.
- Extension de l'utilité de l'interface, par exemple par l'ajout de la touche de raccourci.
- Améliorer les fonctionnalités pour s'adapter aux différents produits de l'industrie.

2.6.3 Réflexions sur l'impact de mon travail

Selon moi, ma contribution permettra aux utilisateurs de simplifier le processus, de réduire le coût d'apprentissage de l'interface et de rendre la visualisation des résultats plus intuitive.

3. Réalisations

Dans la section précédente, j’ai déjà mentionné que mon stage était principalement axé sur **PowerEye**. Dans cette section, je détaillerai mes réalisations, en commençant par l’architecture globale, puis les modules individuels, et enfin certaines fonctionnalités en dehors du corps principal du logiciel.

3.1 Architecture globale de PowerEye

Dans cette partie, je présenterai la composition du **PowerEye** en termes d’architecture global, y compris le choix de modèle de conception, la construction de la structure de données de l’interface et l’apparence de l’interface. De cette façon, vous pouvez d’abord avoir un concept général du **PowerEye**, et comprendre plus facilement la conception du chaque module dans les parties suivant.

3.1.1 Modèle de conception

Le choix du modèle de conception influe grandement sur la manière de développer le programme. Au début du développement du **PowerEye**, j’ai choisi MVVM comme modèle de conception.

Présentation du MVVM

Il existe trois composants principaux dans le modèle MVVM : le modèle, la vue et le modèle de vue. Chacun sert un objectif distinct. La vue est chargée de définir la structure, la disposition et l’apparence de ce que l’utilisateur voit à l’écran. Le modèle de vue implémente des propriétés et des commandes avec lesquelles la vue peut effectuer une liaison aux données. Les classes de modèle sont des classes non visuelles qui encapsulent les données de l’application. Ainsi, le modèle peut être considéré comme une représentation du modèle de domaine de l’application, qui comprend généralement un modèle de données avec une logique métier et une logique de validation.

Application pratique

3.1.2 structure des données

3.1.3 Disposition de l'interface

3.2 Multiples modules pour PowerEye

3.2.1 Préparation des données

3.2.2 Apprentissage automatique

3.2.3 Contrôle de la chaîne d'assemblage

3.2.4 Interroger les données historiques

3.2.5 Configuration des données globales

3.2.6 Flux de travail personnalisés

3.3 Fonctions en dehors du corps principal

3.3.1 Affichage de l'appel de fonction

3.3.2 Exécution de la chaîne de caractères comme un script

4. Conclusion